

ESTRATEGIA DE LOCALIZACIÓN COOPERATIVA PARA SISTEMAS
MULTIROBOT

WILLIAM ELÍAS DÍAZ CASTRO

UNIVERSIDAD DEL VALLE
FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA
PROGRAMA ACADÉMICO DE INGENIERÍA ELECTRÓNICA
SANTIAGO DE CALI
2013

ESTRATEGIA DE LOCALIZACIÓN COOPERATIVA PARA SISTEMAS
MULTIROBOT

WILLIAM ELÍAS DÍAZ CASTRO

Trabajo de grado presentado como requisito para optar por el título de
Ingeniero Electrónico

Directores

Ph.D. EDUARDO F. CAICEDO BRAVO

Ing. ALEJANDRO PUSTOWKA REYES

UNIVERSIDAD DEL VALLE
FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA
PROGRAMA ACADÉMICO DE INGENIERÍA ELECTRÓNICA
SANTIAGO DE CALI

2013

NOTA DE ACEPTACIÓN:

Firma de jurado

Firma de jurado

CONTENIDO

INTRODUCCIÓN	10
1.1 PLANTEAMIENTO DEL PROBLEMA	12
1.2 OBJETIVOS	13
1.2.1 Objetivo General	13
1.2.2 Objetivos Específicos	13
1.3 JUSTIFICACIÓN	14
1.4 ORGANIZACIÓN DEL DOCUMENTO	15
2. LOCALIZACIÓN EN ROBÓTICA MÓVIL	17
2.1 INTRODUCCIÓN	17
2.2 SISTEMAS MULTIROBOT (SMR)	17
2.3 ESTRATEGIAS DE LOCALIZACIÓN	20
2.3.1 Odometría	20
2.3.2 Reconocimiento de marcas	21
2.3.3 Uso de mapas	23
2.4 FILTROS EN LA LOCALIZACIÓN	24
2.4.1 Filtros gaussianos o paramétricos.	24
2.4.2 Filtros no-paramétricos.	27
2.5 LOCALIZACIÓN COOPERATIVA	31
2.6 CONCLUSIONES	35
3. ALGORITMO DE LOCALIZACIÓN INDIVIDUAL	37
3.1 INTRODUCCIÓN	37
3.2 ESTIMACIÓN DE ÁREA DE PROBABILIDAD DE LA POSICIÓN DEL ROBOT	37
3.2.1 Asignación de áreas de probabilidad a posición de marcas.	39
3.2.2 Generación de estimaciones de posición de la referencia M1	41
3.2.3 Generación de estimaciones de posición de la referencia M2	41
3.2.4 Generación de estimaciones de posición de robot	44
3.3 CONCLUSIONES	45

4. ALGORITMO DE LOCALIZACIÓN COOPERATIVA (ELCSM-UV) POR FILTRO DE PARTÍCULAS.....	46
4.1 INTRODUCCIÓN	46
4.2 ESTIMACIÓN DE POSICIÓN E INCERTIDUMBRE DE MANERA COOPERATIVA.....	47
4.2.1 Registro y envío de datos.....	48
4.2.2 Calculo de distancia y orientación entre cada par de robots.....	52
4.2.3 Determinación de incertidumbre de la medida.....	57
4.3 IMPLEMENTACIÓN DE FILTRO DE PARTÍCULAS PARA REDUCIR INCERTIDUMBRE EN ESTIMACIÓN DE POSICIÓN COOPERATIVA.....	59
4.3.1 Algoritmo de remuestreo.....	60
4.3.2 Algoritmo de localización	63
4.4 CONCLUSIONES.....	70
5. PRUEBAS Y RESULTADOS.....	71
5.1 INTRODUCCIÓN	71
- Dispersión de población de partículas.....	72
5.2 ESTIMACIÓN DE POSICIÓN DE UN ROBOT USANDO EL ALGORITMO INDIVIDUAL.....	73
5.3 SIMULACIÓN DEL ALGORITMO ELCSM-UV VARIANDO EL NÚMERO DE PARTÍCULAS	76
5.4 SIMULACIÓN DEL ALGORITMO ELCSM-UV VARIANDO LA POSICIÓN DE INICIO DEL ROBOT OBJETO DE LA ESTIMACIÓN DE LOCALIZACIÓN .	84
5.5 CONCLUSIONES.....	89
CONCLUSIONES	90
TRABAJOS FUTUROS.....	91
BIBLIOGRAFÍA	92
GLOSARIO	99

ÍNDICE DE TABLAS

Tabla 1. Comparación entre técnicas de localización existentes.	36
--	----

ÍNDICE DE ILUSTRACIONES

Figura 1. Tarea de vigilancia: (a) El líder detecta un punto de vigilancia; (b) se dirige al punto y el resto continúa la marcha con un nuevo líder; (c) un robot desea incorporarse a la formación; (d) el robot se incorpora cuando detecta la cola de la formación; (e) el líder detecta un marcador que indica que hay que dividir la formación; (f) se divide en dos sub-formaciones continuando la marcha con sus líderes respectivos.	18
Figura 2. Ilustración de filtros de Kalman: (a) creencia inicial, (b) una medida (en negrilla) con la incertidumbre asociada, (c) creencia después de integrar la medida en la creencia usando el algoritmo del filtro de Kalman, (d) creencia después de un movimiento a la derecha (lo cual introduce incertidumbre), (e) una nueva medida con incertidumbre asociada, y (f) la creencia resultante.....	26
Figura 3. Representación de “partícula” utilizada por los filtros de partículas. En la gráfica inferior derecha se observan muestras dibujadas de una variable aleatoria gaussiana, X . Estas muestras se pasan a través de la función no-lineal mostrada arriba a la derecha de la gráfica. Las muestras resultantes son distribuidas de acuerdo a la variable aleatoria Y	29
Figura 4. (a) Formaciones, (b) técnicas de referencia para la determinación de la posición de la formación y (c) vehículos no tripulados utilizados en las pruebas. .	33
Figura 5. (a) Gráfica más probable cuando ocurren errores en los sensores y (b) gráficas de origami.	34
Figura 6. Robot visualizando marcas en el entorno e incertidumbres asumidas en la medida de la posición de las marcas.....	38
Figura 7. Magnitudes y orientaciones entregadas por el sistema de medida de distancia basado en visión.	40
Figura 8. Área de probabilidad elíptica.....	40
Figura 9. Estimaciones de posición de M1 con respecto al robot Nx1.	41
Figura 10. Datos obtenidos por Nx1 de forma directa e indirecta por parte del robot	

Nx1.....	42
Figura 11. Definición ley de coseno.	42
Figura 12. Generación de estimaciones de posición de M2.	43
Figura 13. Estimación de posición del robot basado en estimaciones de posición de M1 y M2.....	45
Figura 14. Distancias entre robots y marcas, expresadas en coordenadas polares.	50
Figura 15. Escenarios a los que se enfrentan los robots al momento de presentar coincidencias en sus visualizaciones.	51
Figura 16. Vectores generados con referencia al plano de cada robot cuando se visualizan las marcas en común.....	53
Figura 17. Pasos para determinar la distancia relativa entre dos robots.	54
Figura 18. Área de probabilidad estimada por Nx1 al recibir información visualizada por Nx2.	58
Figura 19. Gráfico circular donde están contenidas las probabilidades de las partículas a remuestrear.....	63
Figura 20. Distribución de partículas sobre el área de probabilidad estimada en la sección 4.2.3.	67
Figura 21. Origen de datos utilizados para el cálculo del peso de cada partícula.	67
Figura 22. Media típica de conjunto de creencias y media según observación de Nx2.	69
Figura 23. Posiciones y orientaciones en las que se ubicó el robot para ejecutar el algoritmo de localización individual.	74
Figura 24. Dispersiones y errores de posición generados al ejecutar el algoritmo desarrollado en el capítulo 3.	75
Figura 25. Escenario utilizado para la implementación del algoritmo ELCSM-UV variando el número de partículas.	76
Figura 26. Evolución de la dispersión y el error en cada coordenada del grupo de 10 partículas luego de más de 1500 iteraciones.	77

Figura 27. Evolución de la dispersión y el error en cada coordenada del grupo de 50 partículas luego de más de 1500 iteraciones.	78
Figura 28. Evolución de la dispersión y el error en cada coordenada del grupo de 100 partículas luego de más de 1500 iteraciones.	79
Figura 29. Evolución de la dispersión y el error en cada coordenada del grupo de 500 partículas luego de más de 1500 iteraciones.	80
Figura 30. Evolución de la dispersión y el error en cada coordenada del grupo de 1000 partículas.	81
Figura 31. Evolución de la dispersión y el error en cada coordenada del grupo de 1500 partículas.	82
Figura 32. Evolución de la dispersión y el error en cada coordenada del grupo de 2000 partículas.	83
Figura 33. Robot R2 ubicado en $x = 0$ m, $y = -6$ m y $\theta = 0^\circ$	85
Figura 34. Robot R2 ubicado en $x = -6$ m, $y = -6$ m y $\theta = 0^\circ$	85
Figura 35. Robot R2 ubicado en $x = 11$ m, $y = -6$ m y $\theta = 180^\circ$	85
Figura 36. Dispersión y error para cada coordenada de posición del Robot R2 partiendo de la posición $x = 0$, $y = -6$ y $\theta = 0^\circ$	86
Figura 37. Dispersión y error para cada coordenada de posición del Robot R2 partiendo de la posición $x = -6$, $y = -6$ y $\theta = 0^\circ$	87
Figura 38. Dispersión y error para cada coordenada de posición del Robot R2 partiendo de la posición $x = 11$, $y = -6$ y $\theta = 180^\circ$	88

INTRODUCCIÓN

Un Sistema Multirobot o SMR, en definición, es un sistema formado por dos o más robots que compartiendo un área de trabajo llevan a cabo una tarea asignada. Algunos de los beneficios ofrecidos por esta clase de sistemas comprenden: redundancia, tolerancia a fallos, aumento de la cobertura y del rendimiento, la reconfiguración del sistema es flexible, los sistemas de medición son distribuidos espacialmente, además, el control de un grupo en particular puede ser centralizado o distribuido, garantizando parcialmente la robustez del sistema. Sin embargo, existen problemas que deben ser mejorados para la ejecución exitosa de las tareas encomendadas, entre los problemas más destacados se pueden encontrar, por ejemplo, que la comunicación entre robots puede ser afectada directamente por el ancho de banda del sistema, lo que representa una limitante considerable; por otro lado, se debe fortalecer el trabajo mancomunado entre los sistemas propioceptivos y exteroceptivos; otro problema o reto que se debe superar, es el de diseñar sistemas que cumplan con sus objetivos con una relación costo beneficio aceptable. En general los sistemas multirobot representan un punto de referencia positivo en comparación con los sistemas conformados por un solo robot.

En la robótica móvil se debe dar respuesta a tres preguntas esenciales: ¿Dónde estoy?, ¿A dónde quiero ir? y ¿Cómo llego a ese lugar? [1], la pregunta inicial está enfocada al tema de localización, por lo que, una coordenada de posición con una incertidumbre baja permitirá dar respuesta con una alta confiabilidad a los dos interrogantes restantes, con base en lo anterior se puede inferir que la localización se vuelve una variable crítica en sistemas que hacen uso de robots móviles. Las técnicas de localización más utilizadas en robótica móvil están basadas en: odometría, localización de marcas y reconocimiento de mapas; la localización por

odometría o por propiocepción mide el desplazamiento del robot contando el número de rotaciones de las ruedas del mismo, lo que está sujeto a errores por deslizamientos o por terrenos irregulares; la localización por reconocimiento de marcas se basa en sistemas exteroceptivos, dentro de los cuales se encuentran los sistemas de visión, sonar, laser, GPS, entre otros, que permiten reconocer características naturales o artificiales del entorno para facilitar la tarea de localización del robot mientras este se desplaza [2] [3] [29], la limitación de estos sistemas es que se debe tener contacto obligado con dichas marcas para conocer directa o indirectamente la posición; en los sistemas basados en mapas los robots cuentan con un conocimiento anticipado del entorno donde van a realizar sus actividades, complementando esta información con la recopilada por sus sistemas propioceptivos y/o exteroceptivos, de esta manera se estima cual puede ser la posición con referencia a la información o mapa recibido antes de iniciar operaciones. Las limitaciones en los sistemas basados en mapas se dan cuando se intenta llegar a un lugar desconocido para el robot, ya que, depende necesariamente de la información previa que haya recibido.

Los filtros son métodos estadísticos que permiten estimar el estado siguiente o distribución posterior de un sistema, su fuerte es el de eliminar en cierto grado el ruido que afecta a un sistema, para el caso particular de sistemas de localización en robótica cooperativa, disminuye la incertidumbre generada por los métodos de localización expuestos anteriormente, por ende aumentan la confiabilidad de la medida y de los procesos que estén atados a ello. Dentro de los filtros estadísticos se encuentran: modelo de Markov, Filtro Kalman (KF) y Filtro de Partículas (FP). El modelo de Markov representa el espacio de estados como una matriz[4], siendo entonces un sistema discreto, lineal y multimodal; el filtro de Kalman es un sistema lineal, continuo, unimodal y de alto costo procesal [27]; los filtros de partículas a diferencia del KF son sistemas no lineales, multimodales y continuos [5], [27].

La localización cooperativa es ampliamente utilizada debido a que ofrece la oportunidad de que cada robot virtualmente extienda el rango de acción de sus propios sensores hacia áreas del entorno inalcanzables en un instante de tiempo, lugares de los que no se tendría conocimiento si no es por las observaciones compartidas por parte de otros miembros del sistema cooperativo. El hecho de compartir la información permite aprovechar la redundancia que ofrece un sistema cooperativo, para reducir la incertidumbre generada al estimar las posiciones de cada uno de los miembros cooperantes, todo lo anterior, gracias a que se cuenta con un canal de comunicación que permite llevar a cabo todas las comunicaciones inter-robots. Sin embargo, debido a la dependencia de un canal para la comunicación de todos los robots, se debe trabajar con un límite de información para que de esa manera no se sature el ancho de banda del sistema, además, se debe tener en cuenta que la información que se transmite entre robots, tiene un error implícito, indicando que la certeza de la información no está garantizada,

Es así como en este documento se propone usar información exteroceptiva, utilizando marcas en el entorno, para que un grupo de robots comparta sus observaciones referentes a las marcas, y de esa manera determinen con una incertidumbre implícita la localización del resto de sus compañeros, aplicando métodos estadísticos para reducir dicha incertidumbre y mejorando la confiabilidad del sistema. El sistema será validado utilizando la herramienta software Player/Stage, simulando escenarios que permitan determinar si la incertidumbre asociada a la localización por detección de marcas puede ser reducida.

1.1 PLANTEAMIENTO DEL PROBLEMA

Cuando se trabaja con Sistemas Multirobot (SMR) se busca que dicho sistema tenga la capacidad de entregar una estimación de la posición de cada individuo que lo conforma, con un margen de error mínimo para que las decisiones y

estrategias seleccionadas en las tareas de navegación, sean las más acordes a la situación real a la que se expone el SMR. Para este fin, se debe tener conocimiento de cuál es la posición del SMR con respecto al ambiente a explorar, y además, cada individuo del sistema debe ser consciente de cuál es su posición con respecto al grupo. Las técnicas utilizadas para estimar la localización de robots incluyen, además de la odometría, el uso de balizas o referencias fijas y el uso de mapas preconcebidos u obtenidos mientras se navega, que sirven de punto de apoyo para los robots en el momento de estimar su posición. En el caso específico de un SMR cada elemento se vuelve una fuente importante de información del ambiente que se está explorando, al percibir el entorno mediante la configuración sensorial con el cual ha sido configurado, lo que lo vuelve un sistema redundante, pero también un sistema donde sus fuentes de información agregan incertidumbre a la información recopilada. Es por esto que surge la pregunta: ¿Cómo aprovechar la información redundante en un SMR de manera tal que se logren reducir los errores e incertidumbres asociados a los sistemas de localización de cada robot?

1.2 OBJETIVOS

1.2.1 Objetivo General

Concebir y desarrollar una estrategia de localización que le permita a un grupo cooperativo de robots móviles localizarse en un entorno simulado.

1.2.2 Objetivos Específicos

- Realizar una búsqueda en el estado del arte de sistemas de localización tanto para robots individuales como grupos de robots cooperativos.
- Establecer un algoritmo que permita la localización individual de cada robot

del sistema multirobot (SMR) y establecer su grado de incertidumbre.

- Diseñar y desarrollar una estrategia de localización global que permita disminuir el nivel de incertidumbre de los sistemas de localización individuales de los robots.
- Validar el algoritmo propuesto en entornos simulados.

1.3 JUSTIFICACIÓN

En el mundo se hace una búsqueda exhaustiva por la mejor combinación de algoritmos, mecanismos o sistemas, que permitan determinar con certeza la ubicación de un robot. Con el paso del tiempo estos esfuerzos se han orientado a la robótica cooperativa por su variedad de beneficios, al incrementar el rendimiento, robustez, reducir tiempos y ampliar áreas de cobertura en las tareas de exploración, en comparación a las capacidades de un sistema conformado por un solo robot.

La localización en la robótica es uno de los factores principales que debe ser resuelto para llevar a cabo las tareas que sean asignadas, en este sentido se destacan técnicas basadas en propiocepción, exterocepción y las que combinan las dos anteriores con métodos probabilísticos para aumentar la confianza en la localización. Internacionalmente se han hecho varios aportes importantes en este sentido, logrando resultados importantes que aunque no garantizan resultados exactos, generan confianza y optimismo en este campo de investigación.

En Colombia la robótica está asociada fuertemente a la industria, sin embargo, académicamente se han hecho aportes importantes en robótica cooperativa, como por ejemplo, el trabajo de J. Jiménez y otros en [12], donde trabajan con un grupo de robots en tareas de navegación en ambientes estructurados, por otro lado el proyecto Quemes en [13], que consiste en una plataforma educativa para que

niños puedan programar comportamientos multi-tarea en grupos de robots.

En la Universidad del Valle y específicamente en el grupo de investigación Percepción y Sistemas Inteligentes (PSI), se han hecho cuatro trabajos enfocados en la robótica cooperativa sobre robots reales. En el primero E. Bacca en [11] implementa algoritmos de comportamientos grupales como Homing y Following; en [8] Nini Vélez trabaja sobre el concepto de realidad aumentada en un grupo heterogéneo de robots, uno real y otro virtual, llevando a cabo una tarea de mapeo y posterior navegación cooperativa para llegar a una meta; en [9] Breyner Posso plantea la implementación de un sistema cooperativo de robots móviles para realizar tareas típicas de la aplicación de fútbol robótico; por último, J. Hernández y A. Pustowka en [10] trabajan en la tarea de crear y mantener formaciones por parte de un grupo de robots, los cuales son monitoreados mediante visión global. Esto demuestra el interés que se ha venido despertando en la Universidad del Valle en cuanto al cooperativismo entre entidades robóticas, y aunque falta mucho camino por recorrer este trabajo exploratorio en la localización cooperativa de miembros de un grupo de robots, puede ser utilizado como base para investigaciones futuras, que permitan desarrollar proyectos tanto educativos como comerciales, que aporten al desarrollo de la región.

1.4 ORGANIZACIÓN DEL DOCUMENTO

En el capítulo 2 se hace un análisis a la bibliografía y a trabajos realizados relacionados con: el impacto de los sistemas multirobot en la realización de tareas, ventajas y desventajas que pueden aportar en el trabajo en equipo; las técnicas de localización utilizadas hasta el momento, destacando fortalezas y reconociendo debilidades; además, se analizan los filtros probabilísticos utilizados para mejorar el rendimiento de los sistemas de localización, realizando una breve introducción al principio de funcionamiento; por otro lado se analizan diferentes trabajos donde

se implementan sistemas para solucionar el problema de la localización de manera cooperativa. En el capítulo 3, se determina la incertidumbre generada por un robot, al estimar su posición relativa a dos marcas o referencias establecidas en el ambiente. En el capítulo 4 se expone la propuesta de localización cooperativa *ELCSM-UV*, que permite reducir la incertidumbre del sistema de localización individual expuesto en el capítulo 3. En el capítulo 5 se presentan las pruebas realizadas al algoritmo individual y al cooperativo, estableciendo los criterios de evaluación que permitirán determinar la eficiencia del último. Por último, se presentan conclusiones y trabajos futuros que se vislumbran con base en lo obtenido en este trabajo.

2. LOCALIZACIÓN EN ROBÓTICA MÓVIL

2.1 INTRODUCCIÓN

En este capítulo se hace un estudio de los sistemas multirobot (SMR), de manera que se entienda como estos sistemas pueden potenciar un sistema robótico, al llevar a cabo una tarea considerando sus ventajas y desventajas. También se analizan las diferentes técnicas de localización en la robótica, destacando sus fortalezas y reconociendo sus debilidades. Luego, se hace una descripción de los filtros estadísticos que son utilizados en localización y que al ser fusionados con las técnicas de localización, permiten a la robótica cooperativa disminuir las incertidumbres de las medidas de posición de cada elemento dentro de un grupo de robots, para obtener sistemas que logren realizar tareas con alta confiabilidad.

2.2 SISTEMAS MULTIROBOT (SMR)

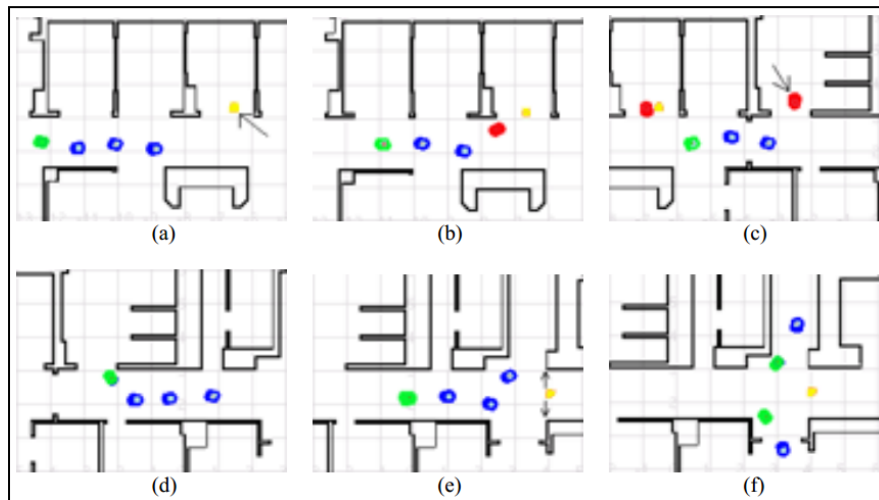
La robótica ha evolucionado con las necesidades del hombre, desde brazos robóticos que evitan al hombre hacer tareas repetitivas, hasta robots móviles exploradores encargados de ingresar en ambientes hostiles para la integridad del cuerpo humano, lo que le dio gran importancia al trabajo con respecto a sistemas que comprendían un único robot. Sin embargo, con el paso del tiempo los sistemas multirobot han tomado una fuerza importante, reflejada en el aumento de esfuerzos en investigación, por sus beneficios y potencialidades comparados con sistemas de un solo robot.

Un sistema multirobot se puede definir como un sistema formado por dos o más robots que pueden actuar de manera cooperativa para alcanzar un objetivo. Los sistemas multirobot permiten realizar actividades que para un único robot serían

demoradas, si no imposibles de llevar a cabo, algunas de esas actividades según C. Agüero y otros en [22] son: Recolección o foraging, Formaciones, Gestión de almacenamiento, Manipulación coordinada, Fútbol robótico, Construcción distribuida de mapas.

Otro ejemplo de tareas realizadas por sistemas multirobots, son los sistemas de vigilancia [15], donde un grupo de robots en formación se desplazan en el ambiente hasta que el líder detecta un punto de vigilancia, entonces rompe la formación y le pasa el rol de líder al robot inmediatamente anterior, como se puede observar en la Figura 1.

Figura 1. Tarea de vigilancia: (a) El líder detecta un punto de vigilancia; (b) se dirige al punto y el resto continúa la marcha con un nuevo líder; (c) un robot desea incorporarse a la formación; (d) el robot se incorpora cuando detecta la cola de la formación; (e) el líder detecta un marcador que indica que hay que dividir la formación; (f) se divide en dos sub-formaciones continuando la marcha con sus líderes respectivos.



E. Bacca en [11] trabaja con un grupo de robots implementando algoritmos de Homing y Following. J. Hernández y A. Pustowka en [10] trabajan con un grupo de

robots en la creación y mantenimiento de formaciones, que pueden ser modificadas al interactuar con obstáculos ubicados en el ambiente; Además, de implementan el concepto de robot virtual para mantener la formación construida.

E. González y otros en [13] hablan sobre las diferentes cualidades y aspectos a tener en cuenta cuando de sistemas multirobot se trata, además, describen como este proyecto se implementa de manera pedagógica en niños para desarrollar habilidades a temprana edad, incentivando la pasión por la robótica y sembrando una semilla de curiosidad en, por qué no, nuevos investigadores.

Algunas de las ventajas que presentan los sistemas multirobot con respecto a los sistemas de un robot individual según J. Jiménez y otros en [12] son:

- Robustez: Si uno de los elementos del grupo robótico falla, puede ser reemplazado por uno de sus compañeros.
- Escalabilidad: Agregar un nuevo elemento al grupo no representa cambios considerables en el sistema.
- Mayor cubrimiento de área: Mayor cantidad de puntos de cobertura sobre un área a explorar.
- Divide y vencerás: La asignación de tareas permite resolver un problema en menos tiempo.
- Simplicidad: Generalmente más simples que los sistemas conformados por un poderoso y único robot, ya que, el potencial del sistema es distribuido entre los componentes del mismo.

Las desventajas de los sistemas multirobot según P. Lima y L. Custodio en [14] son:

- Dificultad para lograr una fusión sensorial eficiente, ya que, la percepción

del mundo viene de diferentes fuentes, con errores implícitos en las observaciones.

- La comunicación es un factor importante a la hora de coordinar las tareas de cada uno de los elementos del grupo robótico, por lo que representa un factor importante a tener en cuenta, debido a que, se pueden presentar problemas de ruido y limitaciones en el ancho de banda.
- La asignación de tareas dentro del grupo debe hacerse teniendo en cuenta factores como capacidades del robot, cercanía al punto donde se debe realizar la tarea, lo que implica planeación, sincronización y comunicación.
- A nivel de arquitectura software, se presentan varios problemas relacionados a ¿cómo y dónde son procesados los datos? y ¿quién toma las decisiones?, estos problemas son abordados usando: arquitecturas centralizadas, descentralizadas y totalmente descentralizadas.

2.3 ESTRATEGIAS DE LOCALIZACIÓN

Las estrategias de localización son utilizadas para garantizar que un robot, o un grupo de robots puedan llevar a cabo tareas asignadas, conociendo su posición con un grado de certidumbre. Las técnicas más utilizadas se basan en:

- Odometría.
- Reconocimiento de marcas.
- Utilización de mapas.

2.3.1 Odometría

La propiocepción en la robótica ayuda a que un robot pueda conocer la posición de su estructura en un momento dado, con base en el estado inicial de sus

sensores, y luego de accionar sus medios de locomoción. Es así como, la odometría en un robot terrestre permite determinar la distancia que se ha desplazado, luego de poner en funcionamiento los motores de sus ruedas, al determinar el número de revoluciones generadas. Se asocian errores de tipo sistemático y estocástico al posicionamiento por odometría, los errores estocásticos se deben a ruido en los sensores mientras los errores sistemáticos se deben principalmente a errores en la descripción del sistema, como por ejemplo cuando los radios de las ruedas del robot no son similares. J. Borenstein y L. Feng en [7], hacen un estudio de los errores sistemáticos para determinar la forma de compensarlos por software, entregando resultados satisfactorios al realizar la evaluación de los mismos. A pesar de lo anterior, se debe tener en cuenta que el error en odometría sin métodos de control, pueden llevar a que se incremente de manera exponencial, algo que no se desea en ningún sistema de localización.

2.3.2 Reconocimiento de marcas

El reconocimiento de marcas está basado en los sistemas exteroceptivos, los cuales permiten al robot ver más allá de su propia estructura. En el caso del hombre, el desplazamiento de la persona *A* hacia un lugar en particular en medio de la ciudad, sin un conocimiento previo, puede volverse una tarea un poco compleja, mientras que, si de manera anticipada la persona *B* le ha indicado a la persona *A*, que características o referencias tiene el entorno por el que se va a desplazar, encontrará que a medida que se desplaza, el ambiente se vuelve familiar, lo que genera confianza en *A* y, además, certeza de que se va a conseguir el objetivo de llegar al lugar deseado. Pues bien, de manera análoga se aplica a los sistemas robóticos al implementarse reconocimiento de marcas en entornos de trabajo, dándole a los métodos de navegación mini-objetivos que permiten a los robots saber dónde están ubicados y en algunos casos, cual es el siguiente paso.

Existen diferentes distribuciones de marcas en el ambiente, por ejemplo, marcas fijas como las utilizadas por G. Zecca en [2] y por J. Woo en [29], donde utilizan referencias con identificación por radio frecuencia (RFID, por sus siglas en inglés), para ayudar al grupo de robots a mantenerse en formación y llevar a cabo actividades planteadas. Otro tipo de configuración es el que utiliza las referencias para marcar los robots del sistema, y por medio de visión administrar sus comportamientos y posiciones, como J. Hernández y A. Pustowka en [10], donde marcan sus robots con referencias de colores para poder controlar las posiciones y orientaciones de los mismos dentro de formaciones, mediante un sistema de visión externo al grupo de robots, además, utilizan el concepto de robot virtual como referencia central para el grupo de robots, facilitando el movimiento del grupo como un solo cuerpo; otro ejemplo de marcas sobre los robots es implementado por L. Coelho en [16] donde controlan la posición y orientación de dirigibles no tripulados, montando sobre estos, figuras geométricas que permiten estimar por visión la posición y orientación de las aeronaves. A. Ramirez-Serrano en [25], utiliza un grupo de robots capaces de medir la distancia entre ellos mediante exterocepción, en este caso los mismos robots operan según el autor como Referencias Dinámicas Inteligentes (IDL, por sus siglas en inglés), al igual que R. Kurazume en [17], donde utiliza una parte del grupo de robots como referencias por un instante de tiempo mientras la otra parte del grupo se desplaza, luego invierten roles para de esa manera alcanzar un objetivo; T. Balch en [23] basa el sistema de localización en el uso del sistema de localización global (GPS, por sus siglas en inglés).

A pesar de las innumerables aplicaciones basadas en la referencia de marcas mediante sistemas exteroceptivos, se presenta la limitación en cuanto a que debe existir una línea directa entre sensores y referencias, por ejemplo la visual para poder ejecutar los algoritmos de localización, es así como incluso el GPS se ve

limitado cuando los equipos a ser localizados se encuentran dentro de estructuras naturales o artificiales, lo que limita el campo de acción de los satélites.

2.3.3 Uso de mapas

El uso de mapas ayuda en algunos casos a que el robot pueda localizarse en un ambiente. Volviendo al ejemplo del hombre, si A desea viajar de un país a otro, y aunque nunca lo ha hecho antes, sale en su vehículo, claro está, con la ayuda de un mapa, que periódicamente revisa para comparar entre lo que puede observar del camino y lo que indica el mapa, de esta manera, su confianza es alta y posiblemente llegue a su destino. De manera análoga, los robots utilizan mapas para comparar la información contenida en ellos con lo que puede ser percibido del entorno. Según [19] la navegación de un robot en un entorno utilizando mapas se puede clasificar en dos tipos:

2.3.3.1 Navegación basada en mapas. Estos sistemas dependen de modelos geométricos creados por el usuario o mapas topológicos del ambiente.

2.3.3.2 Navegación basada en la construcción de mapas. Estos son sistemas que utilizan un sistema sensorial para construir sus propios modelos geométricos o topológicos del ambiente, utilizando el mapa creado para su navegación. Es así como E. Rubión en [18], construye el mapa de un cuarto basado en un láser a bordo del robot y con la ayuda de ultrasonidos para garantizar que su distancia con respecto a las paredes es constante. La anterior es una técnica simple para el mapeo de un lugar a explorar, sin embargo, se utilizan herramientas computacionales más poderosas que permiten estimar un mapa del entorno con una alta fidelidad, claro está, fusionando la información percibida con métodos estadísticos como el Filtro de Kalman el cual permite disminuir la incertidumbre de la medida, como lo hace la técnica de localización y mapeo simultaneo (SLAM, por sus siglas en inglés) utilizada en [19].

2.4 FILTROS EN LA LOCALIZACIÓN

En esta sección se citan conceptos y definiciones expresadas por Sebastian Thrun en [27]. Con el objetivo de brindar al lector, una contextualización con respecto a la familia de filtros utilizados en la localización de robots.

Los filtros utilizan la información del estado anterior de un sistema, la información que dicho sistema puede recopilar del ambiente y una señal de control, todas juntas, para estimar el estado siguiente del sistema. Es así como se destacan dos familias de filtros: los Filtros gaussianos o paramétricos y los filtros no paramétricos.

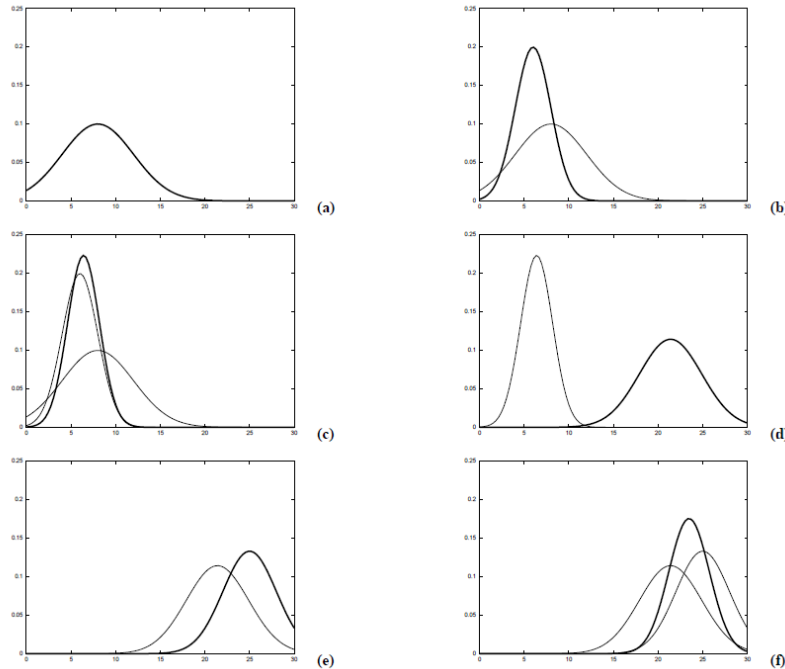
2.4.1 Filtros gaussianos o paramétricos.

El filtro Kalman es la técnica de filtrado más estudiada en los últimos tiempos, inventado para filtrar y predecir en sistemas lineales. El filtro Kalman implementa el cálculo de creencias para estados continuos. No es aplicable a espacios de

estados discretos o híbridos.

En la Figura 2 se muestra el algoritmo de filtro de Kalman para un escenario de localización unidimensional. Suponga que el robot se mueve a lo largo del eje horizontal en cada diagrama en la Figura 2. La creencia anterior del robot es representada como se muestra en Figura 2a. El robot consulta a sus sensores sobre su posición (por ejemplo, un GPS), y estos retornan una medida que es centrada en el pico de la gaussiana en negrilla en la Figura 2b. Esta gaussiana en negrilla ilustra esta medida: Su pico es el valor predicho por los sensores, y su ancho (varianza) corresponde a la incertidumbre en la medida. Combinando la creencia anterior con la medida, mediante el algoritmo de filtro de Kalman, devuelve la gaussiana en negrilla de la Figura 2c. Esta media de la creencia se ubica en medio de las dos medias originales, y su incertidumbre es más pequeña que las de las dos gaussianas contribuyentes.

Figura 2. Ilustración de filtros de Kalman: (a) creencia inicial, (b) una medida (en negrilla) con la incertidumbre asociada, (c) creencia después de integrar la medida en la creencia usando el algoritmo del filtro de Kalman, (d) creencia después de un movimiento a la derecha (lo cual introduce incertidumbre), (e) una nueva medida con incertidumbre asociada, y (f) la creencia resultante.



Fuente. Probabilistic Robotics [27]

A continuación se asume que el robot se desplaza hacia la derecha. Su incertidumbre incrementa debido a que la transición del estado siguiente es estocástica. Esta parte del algoritmo es definida como la predicción o actualización de control, la cual modifica la creencia de acuerdo a una acción, que para el caso de la figura se trata de un desplazamiento, como se observa en la Figura 2d, donde la gaussiana en negrilla representa la predicción hecha por el algoritmo. Luego el robot recibe una segunda medida ilustrada por la gaussiana en negrilla en la Figura 2e, la cual conduce a la creencia posterior en negrilla en la Figura 2f.

En conclusión el paso de actualización de medida decrementa y el de predicción incrementa la incertidumbre en la creencia del robot.

Las asunciones de transiciones de estado lineales y medidas lineales con ruido gaussiano agregado raramente se cumplen, por ejemplo, un robot que se mueve con constante velocidad rotacional y traslacional típicamente se mueve en una trayectoria circular, lo cual no puede ser descrito por transiciones de estado siguiente lineales. El filtro de Kalman Extendido (EKF, por sus siglas en inglés) supera la suposición de linealidad. Aquí el supuesto es que la probabilidad de estado siguiente y las probabilidades de medición se rigen por funciones no lineales.

I. Navarro en [20], utiliza el EKF para reducir la incertidumbre en la localización de un robot móvil generada por las observaciones de marcas distribuidas en el entorno. En [21] se implementa el EKF para realizar una comparación entre la localización medida según el sistema odométrico, y la estimada por el EKF, entregando como resultado que la reducción considerable de las varianzas en cada una de las variables de posición evaluadas.

2.4.2 Filtros no-paramétricos.

Una alternativa popular a las técnicas gaussianas son los filtros no-paramétricos. Los filtros no-paramétricos no dependen de una forma funcional fija de la creencia posterior, como los filtros gaussianos. En lugar de eso, aproximan las creencias posteriores gracias a un número finito de valores, donde cada uno corresponde a una región del espacio de estado. Algunos filtros no-paramétricos dependen de una descomposición del espacio de estado, en el que cada valor corresponde a la probabilidad acumulada de la densidad posterior en una subregión compacta del espacio de estado. Otros aproximan el espacio de estado por muestras aleatorias

dibujadas de la distribución posterior. Si el número de parámetros tiende al infinito, las técnicas no-paramétricas tienden a converger uniformemente a la correcta creencia posterior.

Se tienen entonces dos enfoques no-paramétricos para aproximar creencias continuas sobre espacios continuos con muchos valores finitos. El primero descompone el espacio de estado en muchas regiones finitas, y representa las creencias posteriores con un histograma. Cada región del histograma tiene asignada una probabilidad acumulativa. El segundo enfoque representa las creencias posteriores con muchas muestras finitas. El filtro resultante es conocido como filtro de partículas y ha ganado cierta popularidad en algunos problemas robóticos. Llama la atención que el resultado exitoso de la aplicación de alguna de las dos técnicas, depende en gran medida de la cantidad de partículas o regiones que sean utilizadas, sin embargo, esto afecta fuertemente la carga computacional del procesamiento.

2.4.2.1 Filtros de histograma

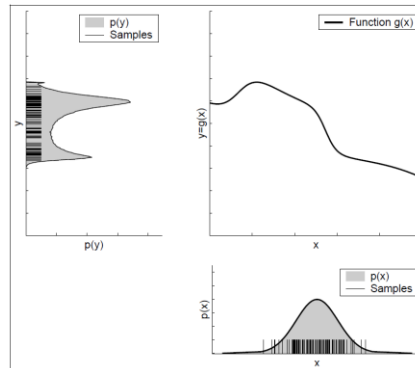
Los filtros de histograma descomponen el espacio de estados en muchas regiones, y representan el acumulado de la creencia posterior para cada región por un solo valor de probabilidad. Cuando se aplican en espacios discretos, se conocen como filtros de Bayes discretos y en espacios de estado continuos, son conocidos como filtros de histogramas.

2.4.2.2 Filtros de partículas

Los filtros de partículas son una alternativa de filtros no-paramétricos, que al igual que los filtros de histogramas aproximan la creencia posterior usando un número finito de parámetros. Sin embargo, difieren en la forma en que estos parámetros

son generados, además de cómo son distribuidos en el espacio de estado. La idea principal de filtro de partículas es la de representar la creencia posterior $bel(x_t)$ por un grupo de muestras de estado aleatorio dibujadas de dicha creencia posterior. La Figura 3 ilustra esta idea para una gaussiana. En lugar de representar la distribución de una forma paramétrica (la función exponencial que define la densidad de una distribución normal), los filtros de partículas representan una distribución con un set de muestras dibujadas a partir de dicha distribución. La representación es aproximada, pero es no-paramétrica, por lo tanto puede representar un espacio mucho más amplio de las distribuciones que, por ejemplo, los filtros gaussianos.

Figura 3. Representación de “partícula” utilizada por los filtros de partículas. En la gráfica inferior derecha se observan muestras dibujadas de una variable aleatoria gaussiana, X . Estas muestras se pasan a través de la función no-lineal mostrada arriba a la derecha de la gráfica. Las muestras resultantes son distribuidas de acuerdo a la variable aleatoria Y .



Fuente. Probabilistic Robotics [27].

A continuación se describe un filtro de partículas básico (ver Algoritmo 1), la entrada corresponde al grupo de partículas X_{t-1} , junto con el control más reciente u_t y la medida más reciente z_t . El algoritmo construye primero un grupo de

partículas temporal \bar{X} que evoca (mas no es equivalente) a la creencia $\overline{bel}(x_t)$.

En el paso 4 del Algoritmo 1 se hace una hipótesis o predicción de estado, $x_t^{[m]}$, que indica la posible posición de la partícula m ubicada en $x_{t-1}^{[m]}$ luego del control u_t , dicho control indica un comportamiento de desplazamiento de la partícula.

En el paso 5 se hace un cálculo del factor de importancia $w_t^{[m]}$, el cual basado en las estimaciones de posición realizadas en el paso anterior, $x_t^{[m]}$, estima la probabilidad de que la medida z_t corresponda a dicha posición.

El punto interesante del filtro de partículas se ve en los pasos 8 – 11, donde se hace un análisis de la relación $\langle x_t^{[m]}, w_t^{[m]} \rangle$, realizando un remuestreo de las partículas con base en su factor de importancia asociado. En este paso se eliminan las partículas con factor de importancia asociado bajo, realizando copias en su lugar de aquellas que poseen factor alto.

Algoritmo 1. Algoritmo básico del filtro de partículas.	
1:	Algorithm Particle filter(X_{t-1}, u_t, z_t):
2:	$\bar{X}_t = X_t = 0$
3:	for $m=1$ to M do
4:	sample $x_t^{[m]} \sim p(x_t u_t, x_{t-1}^{[m]})$
5:	$w_t^{[m]} = p(z_t x_t^{[m]})$
6:	$\bar{X}_t = \bar{X}_t + \langle x_t^{[m]}, w_t^{[m]} \rangle$
7:	endfor
8:	for $m=1$ to M do
9:	draw i with probability $\propto w_t^{[i]}$
10:	add $x_t^{[i]}$ to X_t
11:	endfor
12:	return X_t

Fuente. Probabilistic Robotics [27].

Dentro de la literatura se han encontrado diferentes implementaciones del filtro de partículas, lo cual, indica que se tiene una gran aceptación por su versatilidad al trabajar con sistemas no lineales y su representación multimodal. Por ésta razón es ampliamente usado en localización de robots [28], [29], [30], [31], [32], [33], [34], [35], [36], [37], [38], complementando estimaciones hechas por sistemas de propiocepción, haciendo que sus incertidumbres se vean reducidas y mejorando la eficiencia de los sistemas de localización. En otros como [39], se hacen propuestas para mejorar el algoritmo, modificando el paso de remuestreo al hacer que las partículas con indicadores de importancia bajos se muevan hacia el área donde se encuentran las partículas con mayor indicador.

2.5 LOCALIZACIÓN COOPERATIVA

Los sistemas robóticos cooperativos permiten extender el campo de acción a los mismos miembros del sistema, de tal manera que uno de ellos puede conocer algo

que sucede a pesar de no estar presenciándolo, además, al contar con diferentes fuentes de información existen versiones diferentes de una situación, que al ser compartida puede llevar a reducir incertidumbres y tiempos en las actividades que han sido programadas.

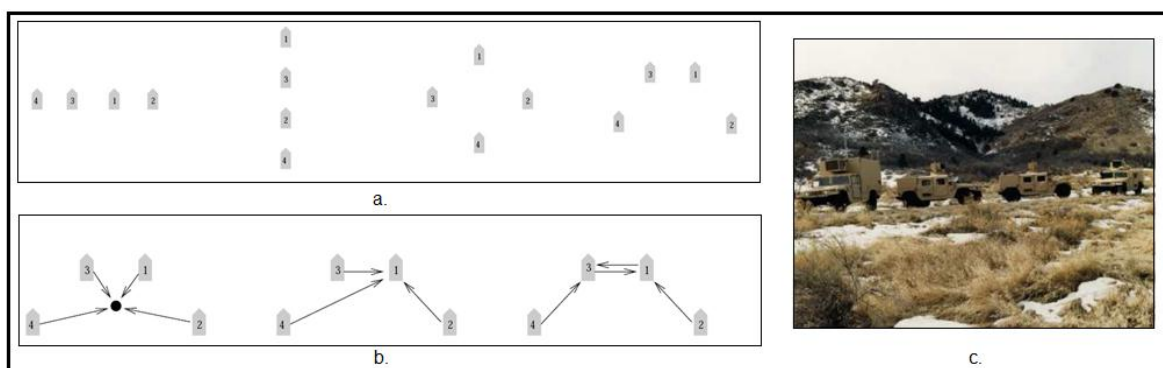
El cooperativismo entre sistemas robóticos es utilizado fuertemente en actividades que implican conocer la localización de los elementos del sistema. Es así como S. Roumeliotis en [40] y [41] formula un KF para determinar la posición y orientación de cada elemento de un grupo de tres robots, donde cada elemento implementa su propio KF para de esa manera mejorar la exactitud en la localización relativa, además de reducir los requerimientos computacionales y de comunicación (proporcionales al número de medidas de posición relativa).

G. Zecca y otros en [2] trabajan con un grupo de cuatro robots que se desplazan aleatoriamente en un ambiente, hasta que encuentran marcas RFID donde encuentran instrucciones para crear formaciones, las cuales son puestas a punto basados en la corrección de su localización, para luego llevar a cabo actividades.

J. Knuth y P. Barooah en [3] trabajan con múltiples robots que de manera cooperativa se desplazan en un entorno, donde cada elemento cuenta con la capacidad de medir las distancias y orientaciones relativas a sus compañeros más cercanos; los autores se basan en la información histórica de posición de cada elemento para determinar su actual posición con base a su odometría, que es confrontada con la información compartida de los sistemas exteroceptivos de sus compañeros para reducir la incertidumbre de su propia posición, para luego actualizar su base de datos. De igual forma Balch y Arkin en [23] usan el Sistema de Posicionamiento Global GPS, para construir y mantener formaciones de vehículos no tripulados que se desplazan a través de campos externos de prueba. En este caso los robots transmiten sus posiciones a los demás miembros de la

formación en coordenadas globales, claro está, con la incertidumbre implícita del GPS del orden de metros y con la imposibilidad de trabajar correctamente en ambientes fuera del alcance del sistema, como por ejemplo, dentro de estructuras naturales y artificiales que impiden la recepción de las señales satelitales. A continuación en la Figura 4a se aprecian las formaciones implementadas en este proyecto, en la Figura 4b se exponen de manera gráfica las técnicas de referencia para determinar la posición de la formación y por último, en la Figura 4c se aprecian los vehículos no tripulados utilizados en las pruebas.

Figura 4. (a) Formaciones, (b) técnicas de referencia para la determinación de la posición de la formación y (c) vehículos no tripulados utilizados en las pruebas.



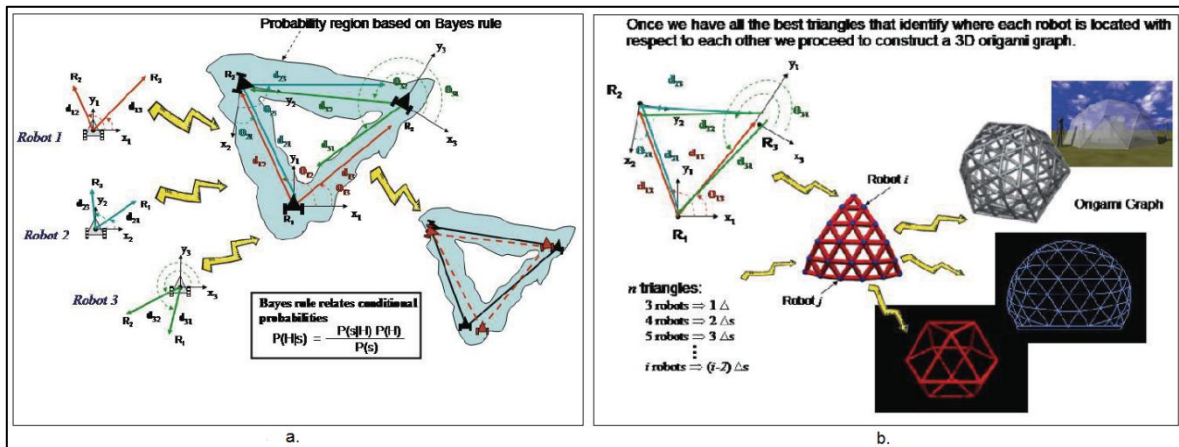
Fuente. Balch T. and Arkin R. C. [23].

Por otro lado, también se ha trabajado con sistemas propioceptivos, como la odometría, base del conocido *dead-reckoning system*, con el problema de que solo presenta resultados aceptables cuando el robot se desplaza distancias cortas, en ambientes en 2D y en condiciones que garanticen un mínimo de errores sistemáticos y no sistemáticos. Sin embargo, al fusionar la odometría con sistemas exteroceptivos se puede obtener una estimación de posición más confiable [24].

En [25] trabajan con un enjambre de robots que en conjunto y basados en las distancias y orientaciones relativas entre ellos, pueden generar lo que los autores

denominan figuras de origami, de esta manera cada vez que se van a desplazar estiman cual va a ser la figura del siguiente estado y al finalizar el desplazamiento comparan las mediciones relativas de distancia entre ellos, para entrar a corregir la posición del o de los robots que se encuentran fuera de la figura de origami esperada. En la Figura 5a se puede apreciar la zona de probabilidad donde pueden estar ubicados los robots que forman el triángulo, en la Figura 5b se muestra la fusión de varios triángulos que forman las llamadas gráficas de origami.

Figura 5. (a) Gráfica más probable cuando ocurren errores en los sensores y (b) gráficas de origami.



Fuente. Ramirez-Serrano A. [25].

Otra forma de aprovechar las bondades de la robótica cooperativa, es utilizándola para detectar objetos ubicados en un entorno, como lo hace C. Agüero en [42] donde se cuenta con un escenario con tres robots con sistemas de visión, donde cada uno con base en su posición actual, indica la ubicación del objeto que se desea encontrar, de manera que se fusionan las tres observaciones para definir un área de localización del objetivo, en consecuencia, con una incertidumbre más baja que las observaciones individuales. De igual forma Sv. Noykov y otros en [43] utilizan referencias fijas en el entorno para determinar cooperativamente y de

manera precisa su posición.

En actividades de mapeo se utiliza el cooperativismo entre robots para confrontar los mapas desarrollados por cada robot integrante del equipo, de esta manera se obtiene una versión mucho mejor del ambiente en lugar de la hecha por cada uno de ellos. Un ejemplo de la utilización del cooperativismo entre dos robots para crear el mapa del entorno y además para localizarse mutuamente es expuesto por D. Fox en [44]. La localización colaborativa es definida como CSLAM por M. Moratuwage y otros en [45], donde utilizan una serie de vehículos marinos para realizar la exploración, mapeo y localización, basados en GPS y sensores laser.

2.6 CONCLUSIONES

En este capítulo se hizo un análisis de las diferentes técnicas de localización desde la más antigua como lo es la odometría, hasta llegar a técnicas que fusionan todas las existentes para sacar el mejor provecho de cada una y obtener la menor incertidumbre posible en la posición de un robot. Una de las técnicas que tiene mucho potencial debido a su posible implementación en muchas de las tareas que se desarrollan en este momento en el mundo, es la que implementa la robótica cooperativa, que toma las técnicas anteriores desarrolladas por un solo robot para mejorarlas al compartir y en consecuencia reducir errores de posicionamiento. En la Tabla 1 se muestra de manera resumida lo encontrado en este capítulo al estudiar las técnicas de localización existentes.

Tabla 1. Comparación entre técnicas de localización existentes.

MÉTODOS DE LOCALIZACIÓN		
Método	Ventajas	Desventajas
Odometría	<ul style="list-style-type: none"> - Simple y bajo costo. - Desplazamientos cortos generan errores pequeños. 	<ul style="list-style-type: none"> - Al ser relativos, producen error acumulativo exponencial.
Reconocimiento de marcas	<ul style="list-style-type: none"> - Permiten reducir la incertidumbre generada por los sistemas de propiocepción. - Sensores con mayor resolución generan medidas con baja incertidumbre. 	<ul style="list-style-type: none"> - No se puede perder el contacto (visual o de comunicación) respecto a las marcas. - Si las marcas son fijas, implica que se debe acondicionar el escenario.
Filtros	<ul style="list-style-type: none"> - Pueden eliminar el ruido de los sistemas, que para el caso de la localización es representado como la incertidumbre de la medida. - Estimación de estado próximo con base en señal de control, mediciones y posiciones actuales. - La carga computacional generada por el Filtro de Partículas depende de la cantidad de partículas implementadas. 	<ul style="list-style-type: none"> - KF solo trabaja en sistemas lineales. - Carga computacional alta para EKF. - Una cantidad baja de partículas para el PF reduce la relación convergencia/tiempo del algoritmo. - Una cantidad alta en el número de partículas representa una alta carga computacional.
Cooperativismo	<ul style="list-style-type: none"> - Mejora en el rendimiento al llevar a cabo una tarea. - Existe percepción distribuida, reduciendo tiempos y ampliando áreas de acción. - Presenta redundancia, haciéndolo tolerante a fallos. 	<ul style="list-style-type: none"> - Limitaciones de comunicación inter-robots por ancho de banda. - Información compartida con errores.

3. ALGORITMO DE LOCALIZACIÓN INDIVIDUAL

3.1 INTRODUCCIÓN

En este capítulo se presenta un algoritmo de localización individual de un robot, con respecto a dos marcas ubicadas en el ambiente en el que se encuentra. En principio se asume que se cuenta con un sistema de visión que permite determinar, con respecto al plano del robot, las distancias y orientaciones a las que se encuentran ubicadas las marcas. Luego se determina la incertidumbre, que de manera implícita viene asociada a la medida de la posición del robot y que servirá de punto de referencia para evaluar el algoritmo de localización cooperativa propuesto en el capítulo 4 de este libro.

3.2 ESTIMACIÓN DE ÁREA DE PROBABILIDAD DE LA POSICIÓN DEL ROBOT

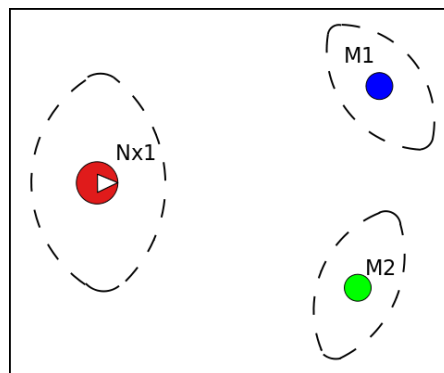
El algoritmo de localización individual se basa en un sistema de visión a bordo del robot, encargado de realizar medidas de distancia con respecto a dos marcas que sirven de referencia al robot. En este punto, se asume que el sistema de visión entrega la distancia y orientación que separan al robot de cada marca, por lo que el algoritmo desarrollado en esta sección, se enfoca en determinar el nivel de incertidumbre de la medida de posición del robot con respecto a dos marcas que sirven de referencia.

Por otro lado, es importante aclarar que este algoritmo se desarrolla con la intención de utilizarlo como la herramienta inicial para el algoritmo de localización cooperativa, que será explicado en el capítulo 4. La finalidad del algoritmo desarrollado en este capítulo se enfoca en generar la distribución de un grupo de

partículas, que en conjunto representan la probabilidad de la posición del robot que lo ejecuta, surge entonces la pregunta: ¿Por qué no trabajar con dispersiones aleatorias o uniformes de las partículas sobre el área donde el robot se desplazará?; la razón que responde a esta pregunta es que con dichas distribuciones se debe generar una gran cantidad de partículas para cubrir el área efectiva de desplazamiento del robot, lo que puede generar una carga computacional alta; mientras que con el algoritmo aquí propuesto se puede estimar un área de distribución mucho menor, sobre la que al final se hace una distribución aleatoria de partículas. Sin embargo, existe el riesgo de que el área estimada por este algoritmo no contenga la posición real del robot que lo ejecuta, riesgo que se corre con la intención de reducir la carga computacional.

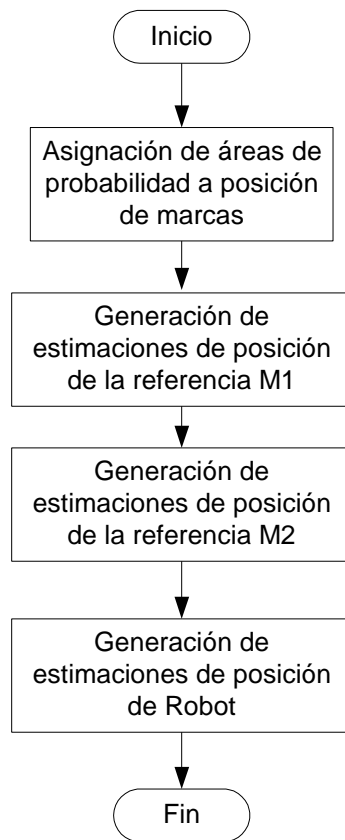
En la Figura 6 se puede apreciar la incertidumbre estimada en la posición del robot Nx1 y las incertidumbres asumidas para las marcas cuando entran en el campo visual del robot.

Figura 6. Robot visualizando marcas en el entorno e incertidumbres asumidas en la medida de la posición de las marcas.



Para determinar el área de probabilidad en la que se estima, puede estar ubicado el robot, se siguen los pasos descritos en el Diagrama 1.

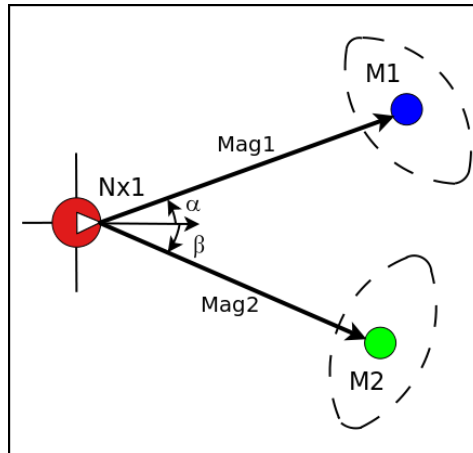
Diagrama 1. Determinación de área de probabilidad de robot en sistema de localización individual.



3.2.1 Asignación de áreas de probabilidad a posición de marcas.

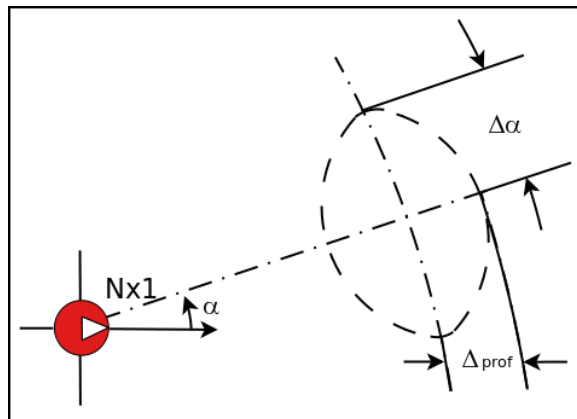
Las medidas entregadas por el sistema de visión hacen referencia a un vector proyectado desde el plano del robot hacia cada una de las referencias observadas, como se muestra en la Figura 7.

Figura 7. Magnitudes y orientaciones entregadas por el sistema de medida de distancia basado en visión.



Con base en las coordenadas de posición de las marcas con respecto al robot, se crean áreas de probabilidad con forma elíptica, como la mostrada en la Figura 8, donde se pueden observar las incertidumbres correspondientes a la posición angular y a la profundidad del objeto visualizado con respecto al plano del robot. El algoritmo permite introducir valores de incertidumbre para cada elipse generada de manera independiente.

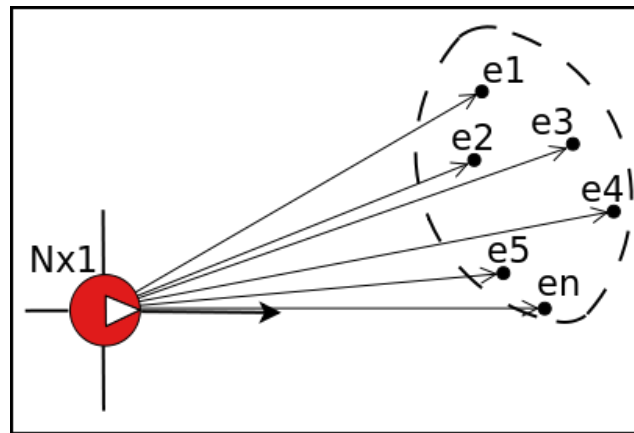
Figura 8. Área de probabilidad elíptica.



3.2.2 Generación de estimaciones de posición de la referencia M1

Es importante aclarar que con M1, se hace referencia a la marca ubicada más a la izquierda del campo de visión de la cámara del robot. Luego de determinar el área de probabilidad, se hace una distribución aleatoria de puntos dentro de dicha área, donde, cada punto hace referencia a una estimación de posición de la marca M1 vista por el robot, como se puede apreciar en la Figura 9. Se generan mil estimaciones ($n=1000$) con la intención de cubrir la mayor parte del área establecida.

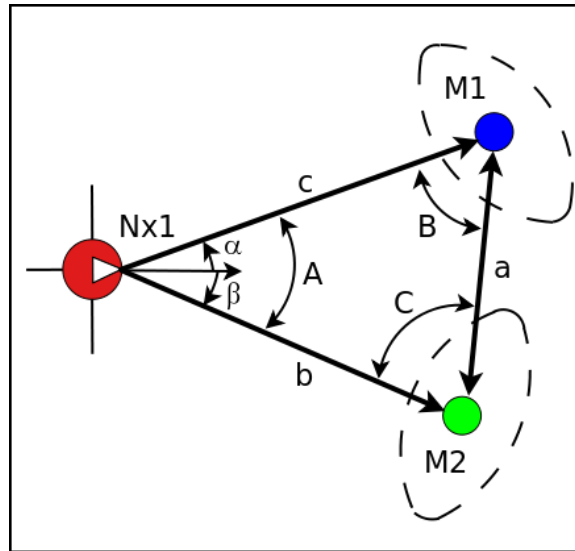
Figura 9. Estimaciones de posición de M1 con respecto al robot Nx1.



3.2.3 Generación de estimaciones de posición de la referencia M2

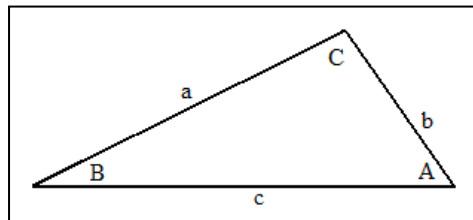
Para determinar las coordenadas de las estimaciones de posición de M2, se deben tener en cuenta los datos obtenidos por el robot Nx1 mediante su sistema de visión, ver Figura 10, en la que se muestran los datos obtenidos de manera directa por el sistema de visión respecto a las marcas visualizadas, $c\angle\alpha$ para M1 y $b\angle\beta$ para M2.

Figura 10. Datos obtenidos por Nx1 de forma directa e indirecta por parte del robot Nx1.



El resto de datos como por ejemplo, la distancia que separa las dos marcas, son calculados de manera indirecta utilizando la ley del coseno, debido a que tanto el robot como las marcas forman un triángulo, ver Figura 11.

Figura 11. Definición ley de coseno.



Las ecuaciones relacionadas al triángulo mostrado en la Figura 11 son:

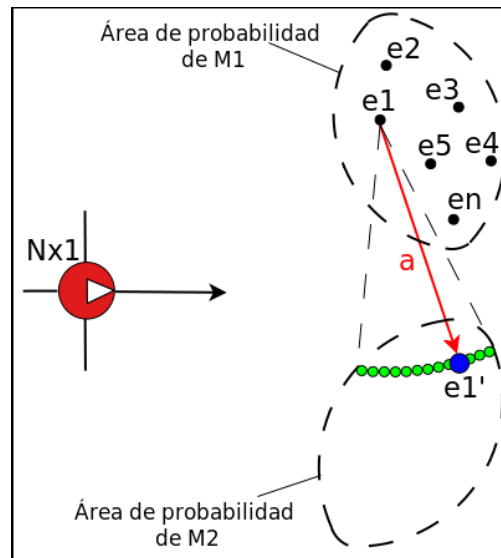
$$a^2 = b^2 + c^2 - 2bc \cos A \quad (1)$$

$$b^2 = a^2 + c^2 - 2ac \cos B \quad (2)$$

$$c^2 = a^2 + b^2 - 2ab \cos C \quad (3)$$

Aplicando la ley de coseno se puede determinar la distancia que separa las dos marcas, con base en las visualizaciones hechas por el robot. Esta distancia es utilizada para encontrar las coordenadas de las estimaciones de posición de M2, realizando una proyección de un vector con origen en la estimación de posición de M1 y dirigida hacia el área de probabilidad de M2 establecida previamente. Al realizar la proyección, se encuentra que existe un número de estimaciones posibles de posición de M2, sobre el arco que dibuja el radio con magnitud a , centrado en la coordenada de la estimación de posición de M1, ver Figura 12.

Figura 12. Generación de estimaciones de posición de M2.



En la Figura 12 se muestra la proyección del vector con magnitud a y origen $e1$ sobre el área de probabilidad de M2, lo que genera una serie de puntos posibles dependientes del ángulo del vector de proyección con origen en $e1$. La selección de la estimación $e1'$ se hace escogiendo de manera aleatoria un ángulo dentro del rango limitado por las líneas punteadas del área de probabilidad de M2. Este

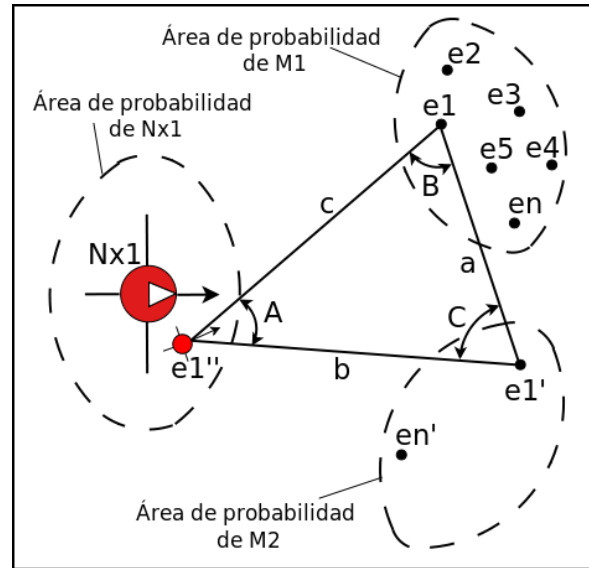
procedimiento se lleva a cabo para cada una de las estimaciones que comprenden el área de probabilidad de posición de M1, generando una pareja correspondiente en el área de probabilidad de posición de M2.

Los casos en que las proyecciones de las estimaciones de M1 sobre el área de probabilidad de M2 no generen alguna estimación, porque la magnitud del vector no alcanza dicha área, son superados reemplazando la estimación de M1 original, por otra estimación dentro del mismo grupo de manera aleatoria y hasta que la estimación seleccionada alcance con su proyección el área de probabilidad de M2.

3.2.4 Generación de estimaciones de posición de robot

Luego de encontrar las estimaciones de las áreas de probabilidad de las marcas, se pueden determinar las estimaciones de posición del robot que las observa. En consecuencia, se toma una pareja de estimaciones de M1 y M2, relacionadas por el método descrito en la sección 3.2.3, a las que se les sobreponen las mediciones directas e indirectas hechas por el robot. En la Figura 13 se muestra como se sobreponen las medidas directas e indirectas mostradas en la Figura 10, sobre las estimaciones $e1$ y $e1'$, generando la estimación $e1''$. Se repite este proceso para cada par de estimaciones de las áreas correspondientes a las marcas, generando el total de estimaciones de la posición del robot $Nx1$.

Figura 13. Estimación de posición del robot basado en estimaciones de posición de M1 y M2.



3.3 CONCLUSIONES

En este capítulo se determinó la incertidumbre de un método de localización de un robot basado en visión, el cual de manera relativa estima su posición con respecto a dos marcas que se encuentran en el ambiente.

Se hace una propuesta de distribución de partículas localizada y luego aleatoria, con la intención de reducir una posible carga computacional a la siguiente etapa, con el riesgo de no incluir dentro del área estimada la posición real del robot que ejecuta el algoritmo.

La incertidumbre asumida sobre la medida de la posición relativa de cada marca, afecta de forma directa la incertidumbre del área estimada donde se encuentra ubicado el robot que ejecuta el algoritmo.

4. ALGORITMO DE LOCALIZACIÓN COOPERATIVA (ELCSM-UV) POR FILTRO DE PARTÍCULAS

4.1 INTRODUCCIÓN

En este capítulo se expone la propuesta de localización cooperativa denominada *Estrategia de Localización Cooperativa de Sistemas Multirobot de la Universidad del Valle, ELCSM-UV*, la cual permite a un robot A estimar la posición de su compañero B , con base en la información exteroceptiva obtenida mediante visión por parte de los robots, al observar dos marcas en común y sin necesidad de que se estén observando entre ellos, dichas marcas están identificadas con un color característico, para que al ser observadas por los robots, se puedan diferenciar. Lo interesante de este algoritmo es que la información recolectada es compartida, logrando así que A estime la posición de B con base en sus propios datos y en los provenientes de B .

Como se comentaba en el capítulo 2, las medidas provenientes de los sistemas exteroceptivos, generan una incertidumbre asociada a la resolución y al tipo de sensor implementado para tal fin. En el caso particular de éste trabajo de grado luego que A calcula la posición exacta de B , se implementa el algoritmo individual en dichas coordenadas para generar el área de probabilidad o incertidumbre correspondiente dicho cálculo, a continuación, se implementa un filtro de partículas al sistema buscando reducir la incertidumbre y generar mayor confianza en la medida final de la posición de B .

El algoritmo se divide básicamente en dos partes, la primera es la encargada de tomar la información obtenida por los dos robots, para estimar la posición de uno con respecto al otro con la incertidumbre implícita que una medición conlleva,

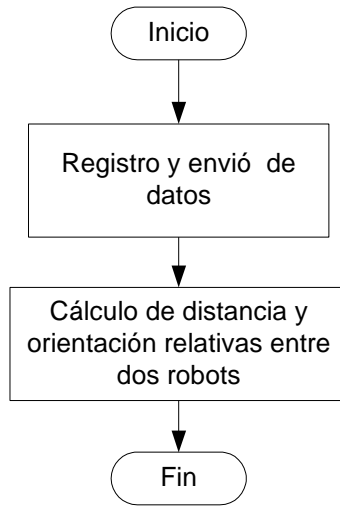
basándose en geometría plana y sin seguir un algoritmo preestablecido de la literatura para tal fin. La segunda parte consiste, en tomar la estimación de posición calculada por la primera parte del algoritmo con su incertidumbre asociada, para implementar un filtro de partículas básico encontrado en la literatura, realizando solo una modificación en su paso de remuestreo propuesta por S. Thrun en [47].

4.2 ESTIMACIÓN DE POSICIÓN E INCERTIDUMBRE DE MANERA COOPERATIVA

Para llevar a cabo el cálculo de la posición se tiene en cuenta la plataforma física y sensorial de cada robot, contando con una cámara ubicada sobre cada estructura, logrando así visualizar los objetos ubicados frente a cada uno de ellos, además, cuentan con ocho sonares ubicados en la parte frontal de su estructura física que combinados permiten realizar las acciones evasivas de obstáculos fijos y móviles. También es importante aclarar que se hace la asunción de que los robots pueden comunicarse entre ellos, de manera que comparten información referente a lo que están observando.

El algoritmo desarrollado estima la distancia y orientación relativas que separan a un par de robots, basándose en información proveniente del sistema de adquisición visual a bordo de cada robot. El algoritmo es ejecutado por uno de los robots, previa recepción de la información que el resto de integrantes del equipo le han enviado. En el Diagrama 2 se muestran las etapas que ejecuta el algoritmo para la consecución del objetivo.

Diagrama 2. Algoritmo para el cálculo de distancia y orientación entre robots.

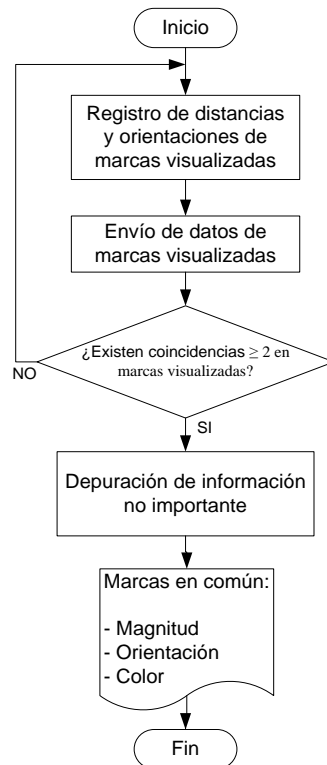


A continuación se explica detalladamente el funcionamiento de cada bloque del Diagrama 2:

4.2.1 Registro y envío de datos.

En esta sección se describe el tipo de información recopilada por cada robot y como ésta es analizada para dar inicio al siguiente estado denominado “Cálculo de distancia y orientación relativas entre dos robots”; en el Diagrama 3 se muestran los sub-estados que comprenden el estado denominado “Registro y envío de datos” seguido de las correspondientes descripciones.

Diagrama 3. Registro y envío de datos

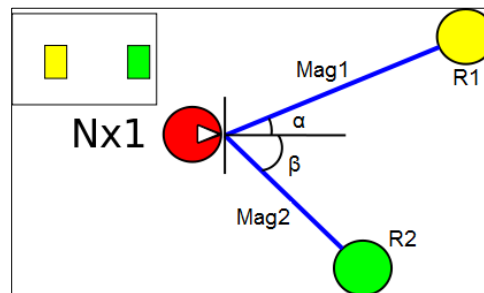


4.2.1.1 Registro de distancias y orientaciones de marcas visualizadas.

Basado en el sistema de adquisición visual, el robot obtiene la distancia, orientación, color y orden en el que son visualizadas las marcas; el orden al que se hace referencia se obtiene de acuerdo a la posición horizontal de cada marca dentro del plano de visión del robot, donde la marca que está más a la izquierda es identificada como la primera en la matriz de datos donde se clasifica la información obtenida. En la Figura 14 se pueden apreciar las componentes polares que corresponden a las distancias entre el robot y dos marcas que se encuentran dentro del campo de visión (arriba a la izquierda de la Figura) de la cámara que se encuentra a bordo, como se puede apreciar, la marca *R1* se encuentra a una distancia *Mag1* con una orientación α , y la marca *R2* se encuentra a una distancia *Mag2* con una orientación β con respecto al plano del

robot. Para el caso de la Figura 14 la marca R1 es identificada como la primer marca por estar más a la izquierda en el campo visual de Nx1.

Figura 14. Distancias entre robots y marcas, expresadas en coordenadas polares.



4.2.1.2 Envío de ID de marcas visualizadas – multicast.

Luego de hacer el registro de datos, los robots envían a los demás miembros del SMR los datos de las marcas visualizadas, que corresponden al orden, magnitud y orientación en que fueron visualizadas.

El condicional “¿Existen coincidencias ≥ 2 en marcas visualizadas?” se encarga de evaluar los datos enviados por el compañero comparándolos con los propios, exactamente, comparando la identificación de las marcas que cada uno ha visualizado, permitiendo el paso al siguiente estado del algoritmo solo si existe una coincidencia de dos o más marcas visualizadas, de otro modo los robots continúan vagando hasta que la coincidencia se dé.

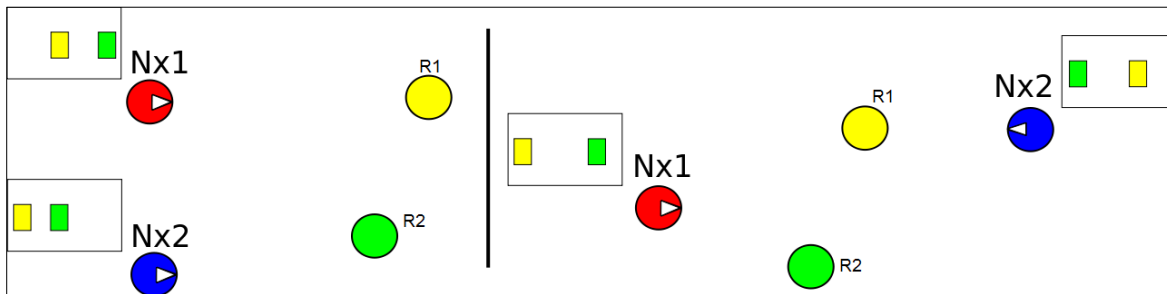
4.2.1.3 Depuración de información no importante.

En esta sección se hace una depuración a las matrices con coincidencias, eliminando aquellas marcas que fueron visualizadas por parte de uno de los dos robots, pero que no contaron con la misma suerte de ser visualizadas por el robot

compañero.

Las distancias obtenidas por $Nx1$ y $Nx2$ son la base para encontrar la distancia y orientación relativa entre dos robots, en la Figura 15 se pueden apreciar los dos escenarios básicos a los que se enfrentan los robots cuando detectan marcas mientras se desplazan por el ambiente.

Figura 15. Escenarios a los que se enfrentan los robots al momento de presentar coincidencias en sus visualizaciones.



Luego de que uno de los robots ha recibido la información de lo que ha visto su compañero, hace un análisis de coincidencia, tomando como variable a comparar el color de las marcas visualizadas. Esto se lleva a cabo por medio del Algoritmo 2.

<i>Algoritmo 2. Detección de coincidencias en marcas visualizadas</i>
<p>Inicializa variables auxiliares $Mx1_{aux} = \{0\}$, $Mx2_{aux} = \{0\}$, $sale = 0$, $cuenta = 0$.</p> <p>Evalúa coincidencias de colores entre matrices de dos robots donde:</p> <p>$MxR[i][2] = color_marca$</p> <p>FOR $i = 0 : 5$</p>

```

FOR  $k = 0:2$ 

  Se evalúa coincidencia con base en el color de marcas observadas.
  IF ( $Mx1[i][2] == Mx2[k][2]$ ) & & ( $Mx1[i][2] \neq 0$ ) & & ( $Mx2[k][2] \neq 0$ )

    Existe coincidencia en una marca observada por los dos robots.
    coincidencias ++

    FOR  $m = 0:3$ 
      Se guarda la información completa de la marca vista por robot 1
       $Mx1_{aux}[i - cuenta][m] = Mx1[i][m]$ 

      Se guarda la información completa de la marca vista por robot 2
       $Mx2_{aux}[k][m] = Mx2[k][m]$ 
      sale = 1

    END FOR

  END IF

  IF (sale == 0)
    cuenta ++
  END IF

  sale = 0

END FOR

END FOR

```

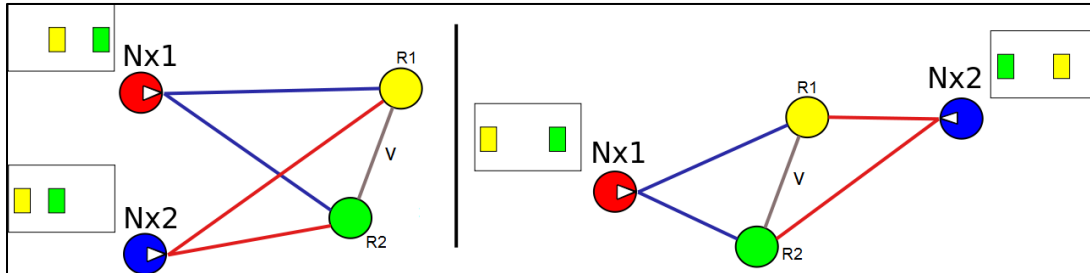
Cada robot entonces, posee dos matrices donde ya se encuentran depurados los datos correspondientes a las marcas visualizadas en común, donde la primera matriz está referenciada al robot que realizará los cálculos y la segunda está referenciada al robot con el que se comparte dicha visualización.

4.2.2 Cálculo de distancia y orientación entre cada par de robots.

Combinando los componentes polares que cada robot genera al observar las

referencias en común y a los escenarios básicos que se pueden generar al realizar el desplazamiento por el ambiente, se presentan las configuraciones de vectores mostradas en la Figura 16.

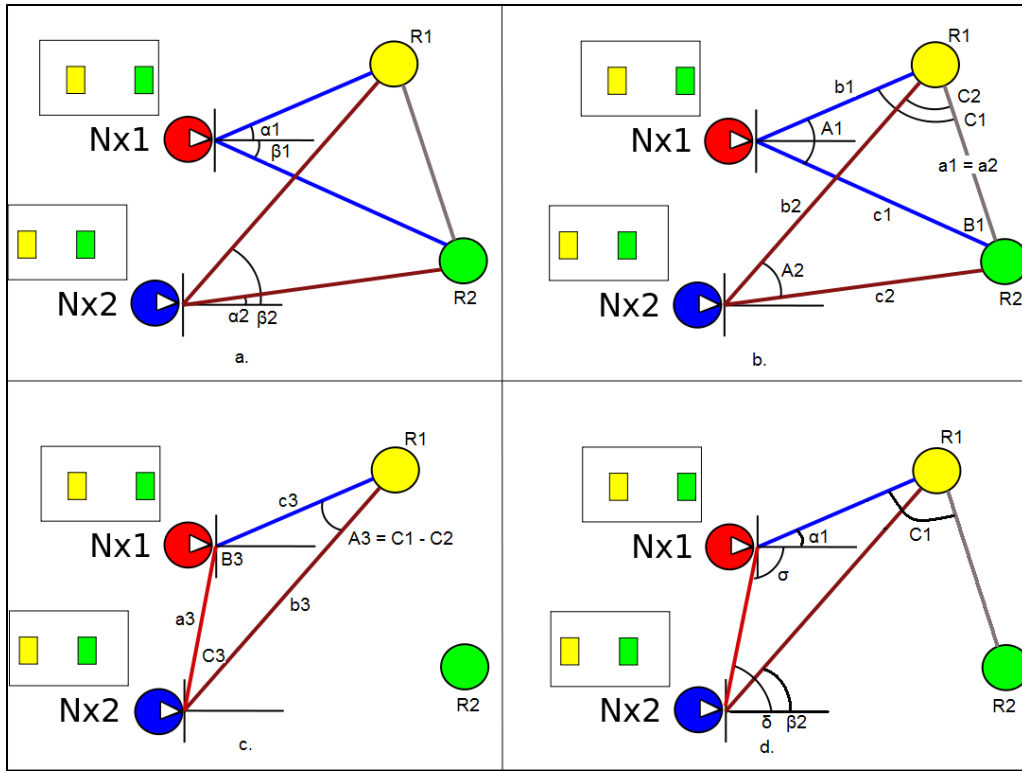
Figura 16. Vectores generados con referencia al plano de cada robot cuando se visualizan las marcas en común.



En la Figura 16 los vectores generados desde $Nx1$ hacia las marcas $R1$ y $R2$ hacen parte de un triángulo cuyo tercer lado es la recta que se traza entre las marcas, identificada como v , de esta manera, se obtienen dos triángulos que comparten el mismo lado, el triángulo generado por $Nx1$ y el generado por $Nx2$, de esta manera se relacionan geoméricamente los dos sistemas sin necesidad de tener una medida directa entre los dos robots.

De acuerdo a la ley de los cosenos se puede calcular el lado de un triángulo opuesto a un ángulo conocido que cuantifica la apertura entre los lados restantes del triángulo también conocidos. Para saber cómo la ley de cosenos puede ayudar a calcular la distancia entre los dos robots, se deben seguir los pasos mostrados en la Figura 17.

Figura 17. Pasos para determinar la distancia relativa entre dos robots.



En primer lugar se toman en cuenta los datos obtenidos por cada uno de los robots relacionándolos en un mismo escenario como lo muestra la Figura 17a, donde se pueden observar claramente los vectores que separan a los robots de cada una de las marcas, además de las aperturas con respecto a sus planos de referencia, también se pueden apreciar los triángulos que se generan entre los robots y las dos marcas. Es aquí entonces, donde entra a operar la ley de los cosenos con el objetivo de obtener el valor numérico de la distancia entre las dos marcas, en ese sentido y a modo de explicación, se basan los cálculos en los datos obtenidos por Nx1. Entonces, si se observa la Figura 17b se tienen los lados del triángulo conocidos b_1 y c_1 , que son los vectores que separan a Nx1 de las marcas R1 y R2 respectivamente, pero falta obtener el ángulo A_1 , que está opuesto al lado del triángulo que se desea encontrar (a_1), éste ángulo se calcula utilizando los ángulos de orientación α_1 y β_1 (Figura 17a),

$$A_1 = \alpha_1 + \beta_1 \quad (4)$$

Despejando la ecuación (1) se aplica ley de coseno para obtener el lado a_1 ,

$$a_1 = \sqrt{b_1^2 + c_1^2 - 2b_1c_1 \cos A_1} \quad (5)$$

Que es el mismo a_2 del triángulo formado por $Nx2$ con las dos marcas; luego de obtener el lado a_1 se procede a calcular los ángulos restantes del triángulo, B_1 y C_1 , despejando las ecuaciones (2) y (3),

$$B_1 = \cos^{-1} \left(\frac{a_1^2 + c_1^2 - b_1^2}{2a_1c_1} \right) \quad (6)$$

$$C_1 = \cos^{-1} \left(\frac{a_1^2 + b_1^2 - c_1^2}{2a_1b_1} \right) \quad (7)$$

De manera similar se procede con el triángulo 2 para hallar los ángulos B_2 y C_2 .

En este momento ya se tiene conocimiento total de los lados y ángulos que componen los dos triángulos generados por los robots y marcas. Al obtener los ángulos C_1 y C_2 se puede calcular uno de los ángulos de apertura de un tercer triángulo que se genera entre la marca $R1$, $Nx1$ y $Nx2$, a este ángulo se le denomina A_3 , ver Figura 17c.

$$A_3 = C_1 - C_2 \quad (8)$$

Se obtiene entonces un nuevo triángulo, triángulo 3, compuesto por los lados:

$$c_3 = b_1 \quad (9)$$

$$b_3 = b_2 \quad (10)$$

Con base en lo anterior, se tienen dos lados conocidos separados por un ángulo de apertura A_3 calculado mediante la ecuación (8), opuesto a este ángulo se encuentra el lado a_3 , cuya magnitud corresponde a la distancia entre los robots que se desea encontrar.

$$a_3 = \sqrt{b_3^2 + c_3^2 - 2b_3c_3 \cos A_3} \quad (11)$$

Se cuenta con los tres lados del triángulo y uno de los ángulos, quedan entonces por calcular los dos ángulos restantes, B_3 y C_3 , que permiten calcular la orientación relativa entre los dos robots.

$$B_3 = \cos^{-1} \left(\frac{a_3^2 + c_3^2 - b_3^2}{2a_3c_3} \right) \quad (12)$$

$$C_3 = \cos^{-1} \left(\frac{a_3^2 + b_3^2 - c_3^2}{2a_3b_3} \right) \quad (13)$$

Ahora se debe determinar la orientación de $Nx2$ con respecto a $Nx1$, para esto se tienen en cuenta los ángulos α_1 , B_3 , β_2 y C_3 , con los que se determinan los ángulos σ (14) y δ (15) véase Figura 17d, donde:

$$\sigma = B_3 - \alpha_1 \quad (14)$$

$$\delta = C_3 - \beta_2 \quad (15)$$

4.2.3 Determinación de incertidumbre de la medida.

Con el algoritmo explicado en la sección 4.2.2 se logra estimar de manera indirecta la posición de otro elemento del SMR con base en información compartida que hace referencia a la distancia y orientación de marcas respecto al robot que las visualiza. Para que cada robot pueda obtener las coordenadas de distancia y orientación en las que se encuentra ubicado uno de sus compañeros, debe llevar a cabo una serie de cálculos, usando herramientas software que de manera implícita agregan errores a las medidas obtenidas. En consecuencia, es importante conocer el error en el resultado después de que se han desarrollado operaciones de tipo aritméticas sobre una medida o sobre el promedio de una serie de valores [26].

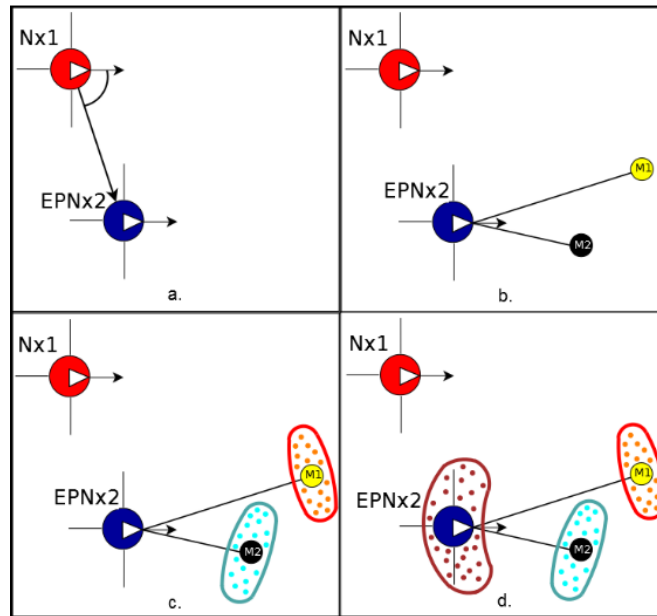
Para el caso particular de éste trabajo de grado, la incertidumbre asociada a la estimación de la distancia y orientación del robot Nx2 con respecto al plano coordenado del robot Nx1, se calcula utilizando la siguiente información:

- Distancia y orientación de Nx2 con respecto al plano de Nx1 obtenidas al ejecutar el algoritmo explicado en 4.2.2.
- Algoritmo de localización individual explicado en el capítulo 3 de éste documento.

En la Figura 18 se puede apreciar cómo se determina el área de probabilidad donde Nx1 cree que está ubicado Nx2, dicha estimación se expresa como EPNx2 (Estimación de Posición de Nx2). En primera instancia Nx1 ubica en su plano coordenado la estimación de posición y orientación de Nx2 obtenidas al ejecutar el algoritmo explicado en 4.2.2, ver Figura 18a. A continuación, Nx1 ubica las marcas

visualizadas en común con Nx2 utilizando los datos enviados por éste último para referirlas a EPNx2, logrando de esa forma que dichas estimaciones de la posición de las marcas queden referidas también a su plano coordenado, ver Figura 18b. Luego de que Nx1 ha ubicado en su plano coordenado las estimaciones de posición del otro robot y de las marcas observadas por este último, se dispone a realizar la estimación del área de probabilidad EPNx2, aplicando el algoritmo explicado en el capítulo 3 de éste documento, inicialmente asigna áreas conocidas de probabilidad a las estimaciones de posición de las marcas, ver Figura 18c, finalmente se obtiene el área de probabilidad de EPNx2 que depende directamente de las dimensiones de las áreas de probabilidad de cada marca, ver Figura 18d.

Figura 18. Área de probabilidad estimada por Nx1 al recibir información visualizada por Nx2.



Se aclara que la estimación del área de probabilidad descrita en ésta sección del documento, depende enteramente de las dimensiones que sean asignadas a las áreas de probabilidad de las marcas visualizadas por los dos robots. En principio,

la estimación obtenida es utilizada para generar condiciones iniciales en la ejecución del método probabilístico que será explicado en la sección 4.3.

4.3 IMPLEMENTACIÓN DE FILTRO DE PARTÍCULAS PARA REDUCIR INCERTIDUMBRE EN ESTIMACIÓN DE POSICIÓN COOPERATIVA

Luego de determinar el área de probabilidad donde uno de los robots cree se encuentra ubicado su compañero de equipo, se busca reducir la incertidumbre generada por el algoritmo descrito en la sección 4.2, para lograr esto se trabaja con filtros de partículas.

La idea principal de los filtros de partículas es la de representar la distribución de densidad de probabilidad posterior del robot, o área de probabilidad, mediante la implementación de un grupo de partículas con peso asociado. El algoritmo trabajado en esta sección se basa en un algoritmo SIR [29] (*Sequential Importance Resampling*), por sus siglas en inglés, el cual, realiza el remuestreo de las partículas eliminando aquellas cuyo peso asociado es relativamente bajo, en comparación al peso de las demás, reemplazando el componente eliminado con una partícula con mayor peso o probabilidad. En [47] se implementa el filtro de partículas SIR, realizando una modificación en el algoritmo de remuestreo, sin dejar de seleccionar las partículas con mejor probabilidad, pero eventualmente, brinda continuidad a partículas con probabilidades muy bajas, llegando al punto de rescatar partículas que en un momento $k - 1$ tenían una probabilidad muy baja, pero en el instante k pueden representar la posición exacta del robot que se desea ubicar. Por otro lado, según [29], [32], [30] y [48], el algoritmo SIR reduce la degeneración, que consiste en que el peso total del grupo de partículas se concentre solo en una de ellas.

4.3.1 Algoritmo de remuestreo

En el Algoritmo 3 se muestra el método de remuestreo denominado “Resampling Wheel” según [47]:

Algoritmo 3: Algoritmo de Remuestreo

1. Inicializa vector auxiliar de partículas

$$p3 = \{0\}$$

2. Inicializa beta y define *random()* :

$$beta = 0$$

random() //Genera valores aleatoriamente entre 0 y 1

3. Se selecciona aleatoriamente uno de los índices del vector de pesos:

$$index = random() * N$$

4. Determinar probabilidad más alta del grupo de partículas:

Función: $mw = \max(w_k^i)$

5. FOR $i = 1 : N$

Sumar: $beta + = random() * 2 * mw$

WHILE $beta > w_{[index]}$

$$beta - = w_{[index]}$$

$$index = index + 1$$

END WHILE

Guardar partícula seleccionada:

$$p3_{[i]} = p_{[index]}$$

END FOR

6. Asignar vector de partículas auxiliares a vector de partículas final:

$$X_k = p3$$

A continuación se hace una descripción más detallada del Algoritmo 3 implementado en el presente trabajo de grado:

El remuestreo consiste en tomar un grupo de partículas con una probabilidad o peso asociado a cada una de ellas y hacer una selección de acuerdo a un criterio en particular para que pasen a una iteración siguiente del filtro de partículas; en la localización se busca seleccionar las partículas que tienen una probabilidad alta, ya que esto indica que dichas partículas se encuentran próximas a la posición real del robot. El algoritmo básico del filtro de partículas sigue al pie de la letra ésta regla, impidiendo que partículas con una probabilidad baja pasen a la siguiente iteración, lo que trunca la posibilidad de tener una partícula en un instante de tiempo con probabilidad baja, pero que en el siguiente ciclo puede representar la posición exacta del robot.

La propuesta del profesor Thrun consiste en representar la suma de todas las probabilidades como un gráfico circular o gráfico de pastel, donde cada sección hace referencia al peso de la partícula representando la probabilidad individual de cada una como lo muestra la Figura 19a (W =peso o probabilidad). De manera gráfica la idea es que el apuntador (flecha negra en la Figura 19b) se desplace angularmente por el perímetro del gráfico circular siguiendo las instrucciones descritas en los paso 3, 4 y 5 del Algoritmo 3, así:

- Paso 3: Se selecciona de forma aleatoria un índice del vector que contiene los pesos o probabilidades de las partículas, de ésta manera se selecciona

la partícula sobre la cual el apuntador (flecha negra) iniciará ver Figura 19b. En éste ejemplo el apuntador inicia en la partícula $P6$ con peso asociado $W6$.

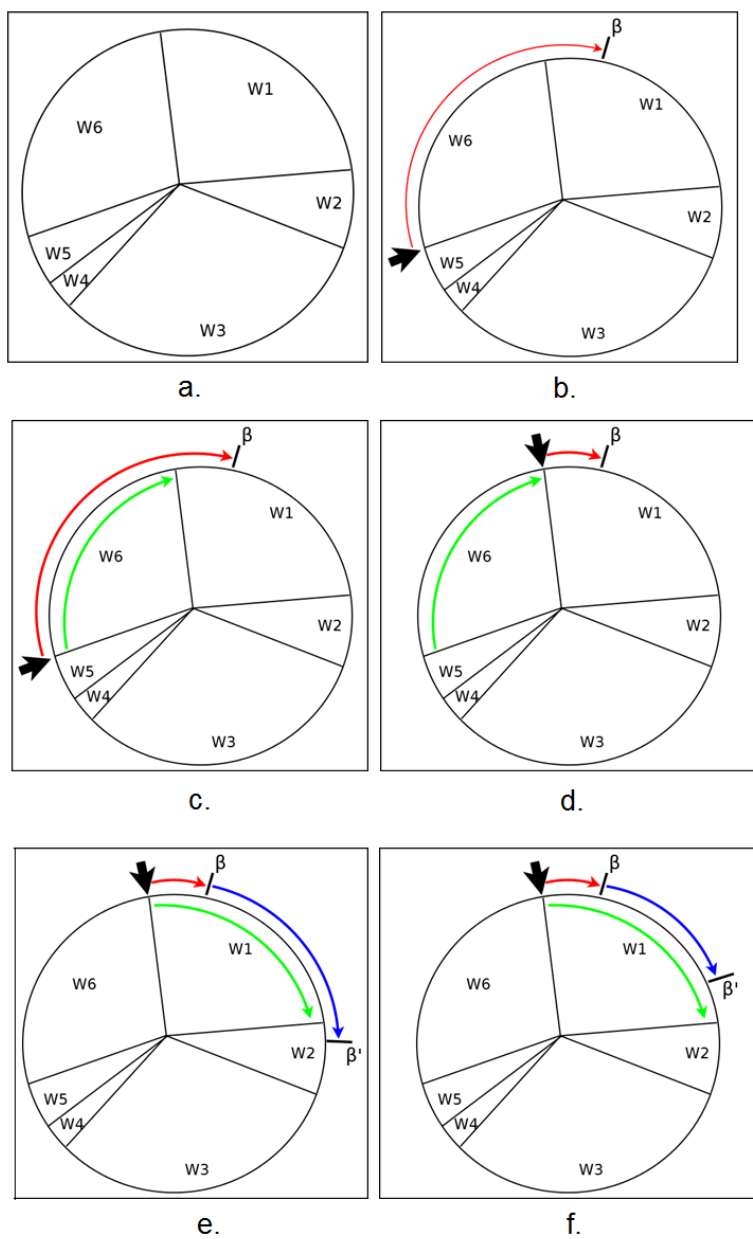
- Paso 4: Se selecciona la probabilidad más alta dentro del grupo de pesos asociados a las partículas.
- Paso 5: A la variable β , inicializada en cero previamente, se le suma un valor aleatorio entre cero y el doble de la probabilidad máxima determinada en el paso anterior.

La variable β es la referencia a ser comparada con el peso de la partícula seleccionada en el paso 3 del Algoritmo 3. Para el ejemplo de la Figura 19 el valor calculado de β resulta ser mayor que $W6$ (flecha verde), ver Figura 19c. Es así como mientras el valor de β sea mayor que el peso de la partícula seleccionada, a β se le restará el peso de dicha partícula (ver Figura 19d) y se aumentará el índice en una unidad haciendo que en el siguiente ciclo del *while* el peso comparado sea el de la siguiente partícula ($W1$), éste paso se repite hasta que se cumpla la condición del ciclo *while*.

Cuando la condición del ciclo *while* no se cumple se agrega un nuevo valor a β dando como resultado un nuevo valor para β , para el caso del ejemplo β' , donde quizá la partícula objeto de comparación presente un peso menor al de β' como se muestra en la Figura 19e o por el contrario un peso mayor que el valor de β' como se muestra en la Figura 19f. En éste último caso, la partícula con peso asociado $W1$ es elegida para pasar a la siguiente etapa del filtro de partículas.

El procedimiento se repite N veces obteniéndose de ésta manera el mismo número de partículas que ingresaron al proceso de remuestreo.

Figura 19. Gráfico circular donde están contenidas las probabilidades de las partículas a remuestrear.



4.3.2 Algoritmo de localización

El algoritmo de localización se divide en dos partes: Definición y Ciclo de trabajo,

donde la definición es la sección donde se establecen parámetros iniciales para el correcto funcionamiento del algoritmo dentro del ciclo de trabajo, donde se genera todo el procesamiento importante.

Con base en lo anterior, se establece un conjunto de partículas sobre el área de probabilidad estimada en la sección 4.2.3, dicho conjunto es representado como $\{X_{0:k}^i, i=1, \dots, N\}$, donde N representa la cantidad de partículas y X_k^i representa las coordenadas sobre un plano, con el ángulo de orientación respectivo, es decir $(x_k, y_k, \text{orientación}_k)$; además, cada partícula tiene un vector de peso asociado $\{w_k^i = 1, \dots, N\}$. La expresión $\{X_{0:k}^i, w_k^i\}_{i=1}^N$ denota una medida aleatoria que caracteriza la distribución de densidad de probabilidad posterior $P\{X_{0:k} | Z_{1:k}\}$.

Inicialmente el conjunto de partículas establecido se ubica de manera aleatoria, es decir, se asigna un valor de manera fortuita a cada componente de posición de las partículas, de esta manera se cubre el área estimada. Además, se asigna un peso inicial a cada partícula que establece la probabilidad de que la posición de la partícula, sea la misma del robot al cual se intenta estimar su posición, por ser la primera asignación se hace de manera uniforme, $w_k^i = 1/N$. Luego de establecidas las condiciones iniciales el algoritmo de localización es el siguiente:

<p>Algoritmo 4: Algoritmo de localización</p>

<p>1. Inicializa variable de peso acumulado</p>

$\eta = 0$

<p>2. Remuestreo: $P(X_k Z_k) \sim P(Z_k X_k)P(X_k)$</p>

<p>Trabaja el Algoritmo 3.</p>

3. Predicción

```
FOR  $i = 1:N$   
     $P(X_k^i) = \sum P(X_k^i | X_{k-1}^i) P(X_{k-1}^i)$   
END FOR
```

4. Cálculo de pesos

```
FOR  $i = 1:N$   
     $w_k^i = P(Z_k | X_k^i)$   
     $\eta = \eta + w_k^i$   
END FOR
```

5. Normalización

```
FOR  $i = 1:N$   
     $w^i = \frac{1}{\eta} w_k^i$   
END FOR
```

6. Saltar a *Remuestreo* y repetir cíclicamente

En el Algoritmo 4, es claro que el primer remuestreo no va a tener efecto, debido a que en ese instante no hay diferencia entre los pesos de cada partícula, lo que va a ser diferente en el siguiente ciclo de trabajo.

Paso 1: Se prepara la variable η para guardar el acumulado de la suma del peso de cada partícula.

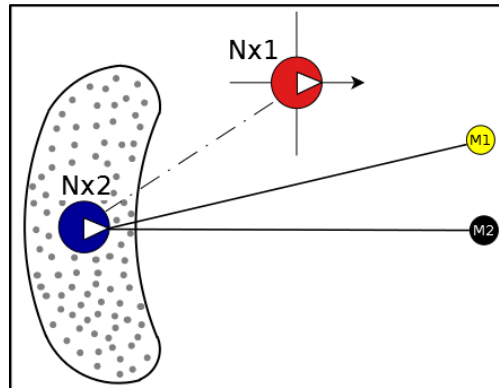
Paso 2: Se realiza el remuestreo de las partículas que como se mencionaba antes para el primer ciclo no es significativo. En la expresión $P(X_k | Z_k) \sim P(Z_k | X_k) P(X_k)$, $P(Z_k | X_k)$ representa la probabilidad de la medición Z_k dada la posición de la partícula X_k en el tiempo k , también

denominada peso de importancia; por otro lado $P(X_k)$ representa las coordenadas previas de cada partícula; por último $P(X_k | Z_k)$ puede representar las partículas resultantes o posteriores, y en conjunto, generan el área de probabilidad donde sería posible estimar la posición del robot que se desea ubicar. Este paso se rige básicamente por la *Regla de Bayes*.

Paso 3: Se predice la posición de las partículas luego del desplazamiento, tanto de los robots como de las partículas. La expresión $P(X_k^i) = \sum P(X_k^i | X_{k-1}^i)P(X_{k-1}^i)$ está compuesta por $\sum P(X_k^i | X_{k-1}^i)$ que representa el modelo de desplazamiento de las partículas, en este caso es gaussiano debido a que va acompañado por una señal de ruido, agregándole una incertidumbre a la medida.

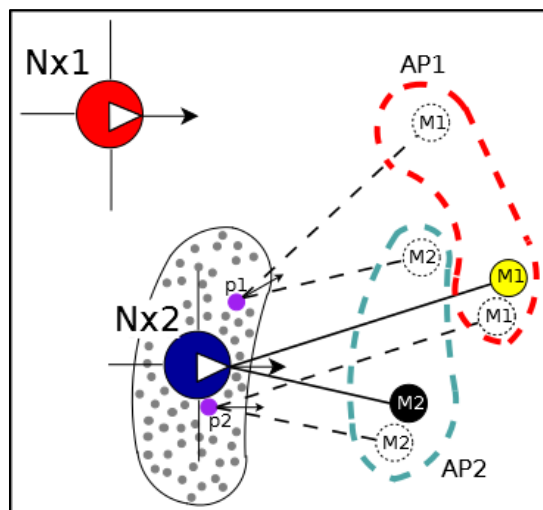
Paso 4: Luego de realizar el desplazamiento de las partículas se hace el cálculo de los pesos de cada una, es decir, se determina la probabilidad de que cada partícula se encuentre ubicada cerca a las coordenadas del robot que se intenta localizar con respecto al robot que realiza la búsqueda. En la Figura 20 se observa un ejemplo de la distribución de las partículas sobre el área de probabilidad estimada en la sección 4.2.3, tal distribución es una de las condiciones para ejecutar el Algoritmo 4. En este caso, el robot Nx1 estima el área de probabilidad del lugar donde puede estar ubicado el robot Nx2 y luego lanza las partículas para aplicar el algoritmo del filtro de partículas con la intención de reducir el área de probabilidad inicial, en conclusión, tener mayor certeza en la estimación.

Figura 20. Distribución de partículas sobre el área de probabilidad estimada en la sección 4.2.3.



El peso de cada partícula se calcula teniendo en cuenta la información que $Nx1$ recibe de $Nx2$, específicamente la distancia y orientación que separa a $Nx2$ de las marcas observadas por los dos robots en el mismo instante de tiempo, como se puede observar en la Figura 21, donde $Nx2$ con respecto a su plano proyecta vectores (línea continua) hacia las marcas observadas $M1$ (Amarilla) y $M2$ (Negra).

Figura 21. Origen de datos utilizados para el cálculo del peso de cada partícula.

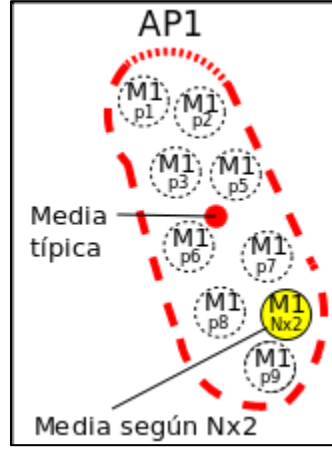


Estos vectores, magnitud y orientación, son sobrepuestos en el plano de

coordenadas de cada partícula, como se puede apreciar en la Figura 21, en las coordenadas de la partícula 1 (p_1) se establecen los vectores que Nx_2 le entregó a Nx_1 , pero que debido a su orientación, indican que las marcas se encuentran ubicadas en lugares diferentes a los reales, estas creencias están representadas por círculos con líneas discontinuas; en el caso de la partícula 2 (p_2), al tener una orientación similar a la de Nx_2 , pero coordenadas x e y diferentes, presenta una creencia muy cercana a las verdaderas posiciones de las marcas reales vistas por Nx_2 . Ahora, si se repite el procedimiento para cada partícula, se obtendrán creencias de la posición de cada marca por parte de cada partícula, generando para M_1 el área de probabilidad AP_1 , y AP_2 para el caso de M_2 . El objetivo es calcular dos probabilidades que indiquen, que tan cerca se encuentran las creencias de la posición de M_1 y M_2 con respecto a lo medido por Nx_2 . En conclusión se obtienen dos probabilidades por partícula, cuyo producto determina que tan cerca está dicha partícula de la posición real de Nx_2 .

La Figura 22 ofrece una vista más detallada del área de probabilidad AP_1 mostrada en la Figura 21, en este caso se cuenta con un número mayor de creencias estimadas por diferentes partículas. La distribución normal de un grupo de muestras (creencias) genera una media típica representada en este caso por un punto rojo, para el caso particular la media es el dato entregado por Nx_2 , es decir, la medida de posición de M_1 según Nx_2 es la media para el área de probabilidad AP_1 . Con base en lo anterior se realiza el cálculo de la probabilidad o peso de las creencias, tanto de M_1 como de M_2 por parte de cada partícula, finalmente, el producto de los dos pesos calculados determina el peso de cada partícula.

Figura 22. Media típica de conjunto de creencias y media según observación de Nx2.



El cálculo del peso de cada partícula se hace a partir de la ecuación de variable gaussiana bi-valuada, caso particular de variable bidimensional [49], aplicando dicha ecuación para cada grupo de creencias de M1 y M2.

$$f_{xy}(x, y) = \frac{1}{\sigma_x \sigma_y 2\pi \sqrt{1 - \rho^2}} \exp\left(-\frac{1}{2(1 - \rho^2)} z\right) \quad (16)$$

$$z = \frac{(x - \eta_x)^2}{\sigma_x^2} - 2\rho \frac{(x - \eta_x)(y - \eta_y)}{\sigma_x \sigma_y} + \frac{(y - \eta_y)^2}{\sigma_y^2} \quad (17)$$

Donde x e y hacen referencia a las coordenadas de posición de cada partícula en el plano; σ_x y σ_y representan las desviaciones de X e Y respectivamente; η_x y η_y representan la media de cada variable; ρ se define como el coeficiente de correlación entre las variables X e Y .

$$\rho = \frac{C_{xy}}{\sigma_x \sigma_y} = \frac{E\{(X - \eta_x)(Y - \eta_y)\}}{\sigma_x \sigma_y} \quad (18)$$

C_{XY} es la covarianza del sistema.

Paso 5: Se realiza la normalización del vector de pesos calculados en el paso anterior, donde se determinó el acumulado con la expresión $\eta = \eta + w_k^i$. Luego se aplica $w^i = \frac{1}{\eta} w^i$ a cada peso para obtener el valor normalizado.

Paso 6: Se salta al paso 2, repitiendo de manera cíclica.

4.4 CONCLUSIONES

En este capítulo se desarrolla el algoritmo para la estrategia de localización cooperativa denominada *ELCSM-UV*, este algoritmo se encuentra dividido en dos partes, que básicamente corresponden a: la estimación del área de probabilidad donde se estima que se encuentra ubicado el robot compañero de equipo y el algoritmo de localización cooperativa, el cual consiste en tomar como entrada el área de probabilidad generada en la primera parte del algoritmo, para reducir la incertidumbre de la medida, aplicando filtro de partículas.

5. PRUEBAS Y RESULTADOS

5.1 INTRODUCCIÓN

Para la validación del algoritmo propuesto se diseñaron pruebas en un entorno simulado, con la intención de obtener indicadores que permitan concluir si se logra una reducción de la incertidumbre en la localización del sistema. A continuación se definen las pruebas realizadas:

1 Estimación de posición de un robot usando el algoritmo individual.

Se ejecuta el algoritmo desarrollado en el capítulo 3 sobre un robot en un ambiente simulado, junto a este se ubican dos marcas según necesidades propias del algoritmo. El robot no presenta desplazamiento y es ubicado en diferentes coordenadas entorno a las marcas ya establecidas, haciendo que estas siempre estén dentro del campo visual del robot. El procedimiento se repite para cada coordenada, obteniéndose la dispersión de las partículas que representan la estimación de posición del robot con respecto a las marcas observadas.

2 Implementación del algoritmo *ELCSM-UV* variando el número de partículas.

Para comprobar la relación entre la convergencia del filtro de partículas y el número de partículas implementadas, se posicionan un par de robots en un ambiente común para que realicen un desplazamiento de modo aleatorio, haciendo que uno de los robots ejecute el algoritmo desarrollado en el capítulo 4. La dispersión de los datos y el error relativo con respecto a la posición real del robot de cada prueba en donde se varía el número de partículas, permiten estimar la relación convergencia/No. de partículas implementadas. Además, es

conveniente conocer cuál es número de partículas que permita reducir la carga computacional y entregue resultados con menos error relativo y baja dispersión.

3 Simulación del algoritmo *ELCSM-UV* variando la posición de inicio del robot objeto de la estimación de localización.

Luego de determinar el número de partículas que permite equilibrar la carga computacional, el error relativo y la dispersión de los datos. Se posicionan dos robots dentro de un ambiente simulado en torno a dos marcas visuales, para que uno de ellos ejecute el algoritmo desarrollado en el capítulo 4; las posiciones de partida de los robots son cambiadas en cada iteración, con la intención de ver el comportamiento de la dispersión de los datos y el error relativo.

Teniendo en cuenta las pruebas diseñadas, los parámetros a evaluar son:

- Dispersión de población de partículas.

Como una medida de la dispersión de la población de partículas generada por los algoritmos, se calcula la desviación estándar poblacional para cada una de las coordenadas de posición (x , y y $Theta$). La ecuación (19) corresponde a la desviación estándar calculada para cada una de las coordenadas de posición.

$$\sigma_a = \sqrt{\frac{\sum_{i=1}^N (a_i - \mu_a)^2}{N}} \quad (19)$$

donde:

σ_a es desviación estándar del grupo de datos de la coordenada a .

a_i es el i-ésimo dato del grupo de datos de la coordenada de posición a .

μa es la media del grupo de datos de la coordenada de posición.

N es el número total de datos dentro del grupo.

- Error relativo de posición y orientación:

Se evalúa el error relativo entre la media de cada una de las coordenadas del grupo de partículas con respecto a la posición real del robot. El error es calculado como lo indica la ecuación (20)

$$e_a = |p_a - \mu_a| \quad (20)$$

donde:

e_a es el error relativo en cada coordenada con respecto a la media.

p_a es la coordenada de la posición real del robot.

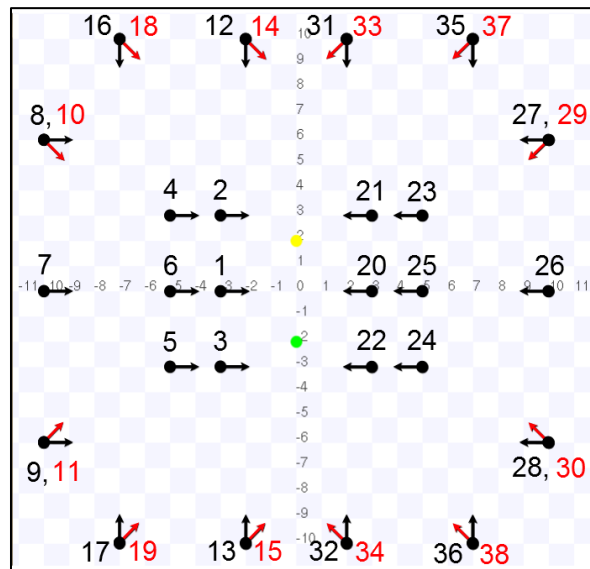
La posición real del robot se obtiene usando la clase *SimulationProxy* del cliente Player, mediante la interfaz *Simulation* y específicamente con la función *GetPose2D*. Ésta función tiene como único requerimiento el nombre del objeto simulado al cual se le desea conocer la posición y orientación.

5.2 ESTIMACIÓN DE POSICIÓN DE UN ROBOT USANDO EL ALGORITMO INDIVIDUAL

En la Figura 23 se pueden apreciar las coordenadas donde es posicionado el robot que ejecuta el algoritmo individual desarrollado en el capítulo 3, además, también se observan las marcas M1 y M2 que sirven de referencias visuales para el robot. Para claridad del lector, en la Figura 23 los puntos hacen referencia a la posición x e y mientras que las flechas indican la orientación en cada prueba, las

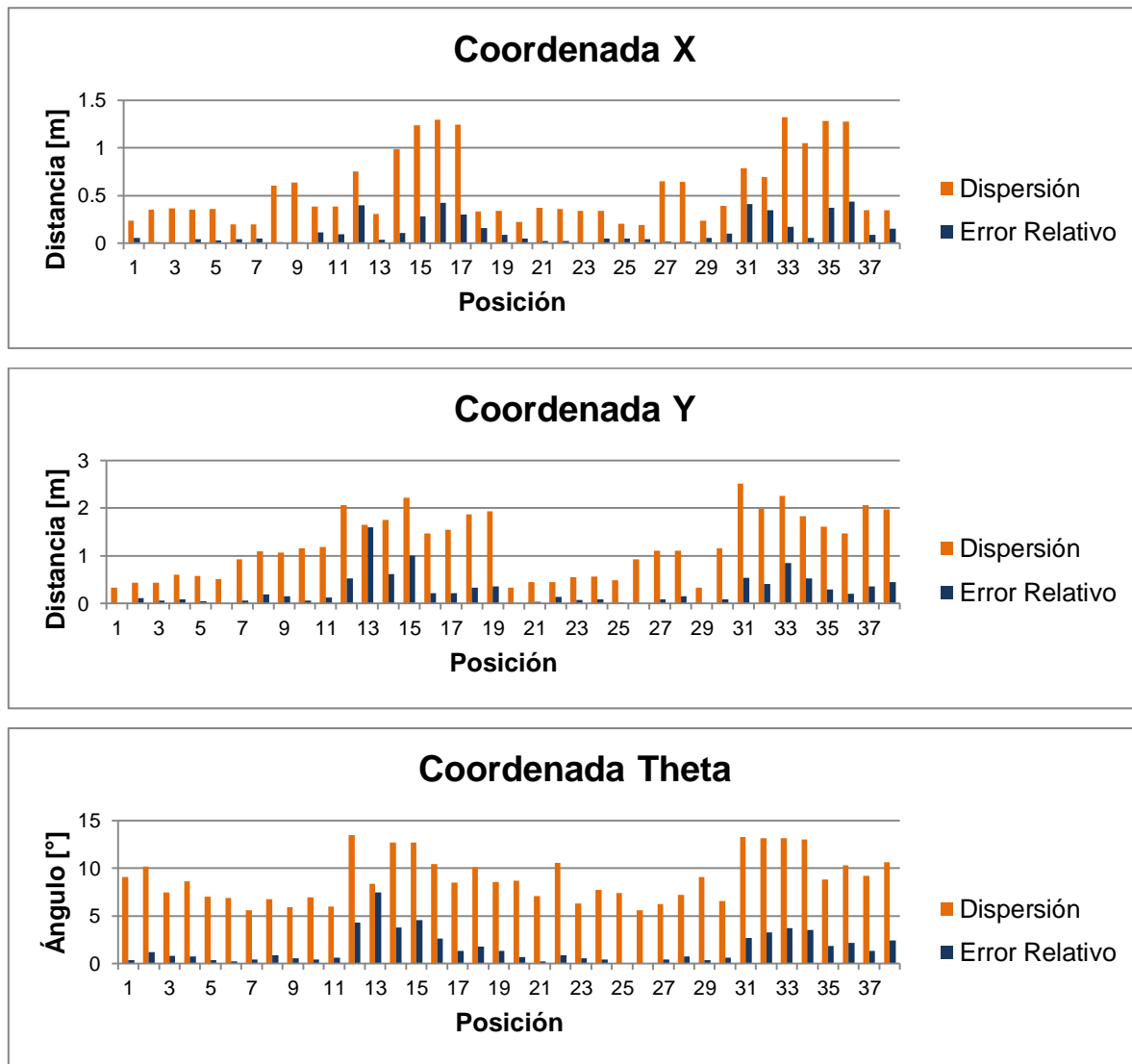
pruebas fueron identificadas numéricamente del 1 al 38 y en los casos donde se hicieron dos pruebas con la misma posición pero con diferente orientación, se diferenciaron cada prueba con colores para su posterior análisis en las gráficas de datos obtenidos.

Figura 23. Posiciones y orientaciones en las que se ubicó el robot para ejecutar el algoritmo de localización individual.



En la Figura 24 se presentan la dispersión y el error generados al ejecutar el algoritmo individual en cada una de las posiciones indicadas en la Figura 23. Como se puede apreciar, las dispersiones y errores de mayor magnitud se presentan en aquellas posiciones donde el ángulo generado por los vectores imaginarios que parten desde el robot hacia cada una de las marcas es menor.

Figura 24. Dispersiones y errores de posición generados al ejecutar el algoritmo desarrollado en el capítulo 3.

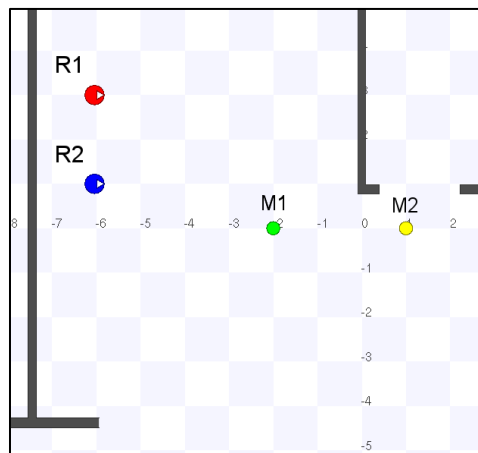


De acuerdo a los resultados de la caracterización del algoritmo uno mostrados en la Figura 24, en las posiciones 12 a 19 y 31 a 38 (ver Figura 23) se presenta la mayor dispersión de datos.

5.3 SIMULACIÓN DEL ALGORITMO ELCSM-UV VARIANDO EL NÚMERO DE PARTÍCULAS

Se posicionan dos robots y dos marcas como se observa en la Figura 25, haciendo que el robot R1 ejecute el algoritmo desarrollado en el capítulo 4. El algoritmo estima un área de probabilidad para la posición del robot R2 con base en el algoritmo individual; dicha área se convierte en el grupo inicial de partículas a ser modificado por el filtro de partículas, que apoyado en las observaciones del propio robot R2 reducirá tanto la dispersión de las partículas, como el error relativo de posición para cada coordenada. El número de partículas seleccionadas para un total de siete pruebas fueron: 10, 50, 100, 500, 1000, 1500 y 2000. Se presentan gráficas de dispersión y error absoluto de las coordenadas X , Y y θ en valores porcentuales con respecto a los valores iniciales de la iteración uno.

Figura 25. Escenario utilizado para la implementación del algoritmo ELCSM-UV variando el número de partículas.



De la Figura 26 a la Figura 32 se muestran los resultados obtenidos luego de simular dos robots después de 1500 iteraciones en promedio para cada prueba.

Figura 26. Evolución de la dispersión y el error en cada coordenada del grupo de 10 partículas luego de más de 1500 iteraciones.

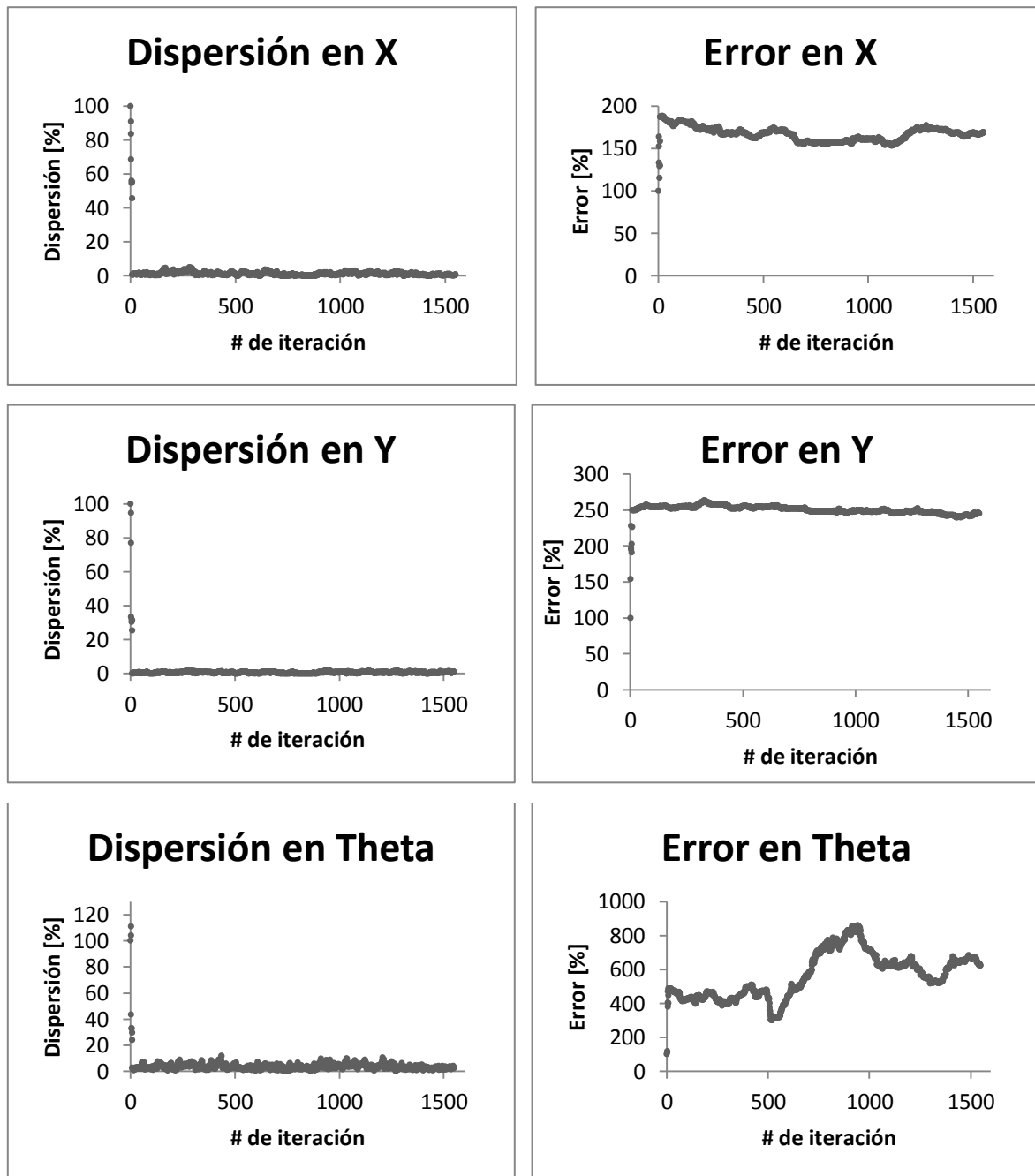


Figura 27. Evolución de la dispersión y el error en cada coordenada del grupo de 50 partículas luego de más de 1500 iteraciones.

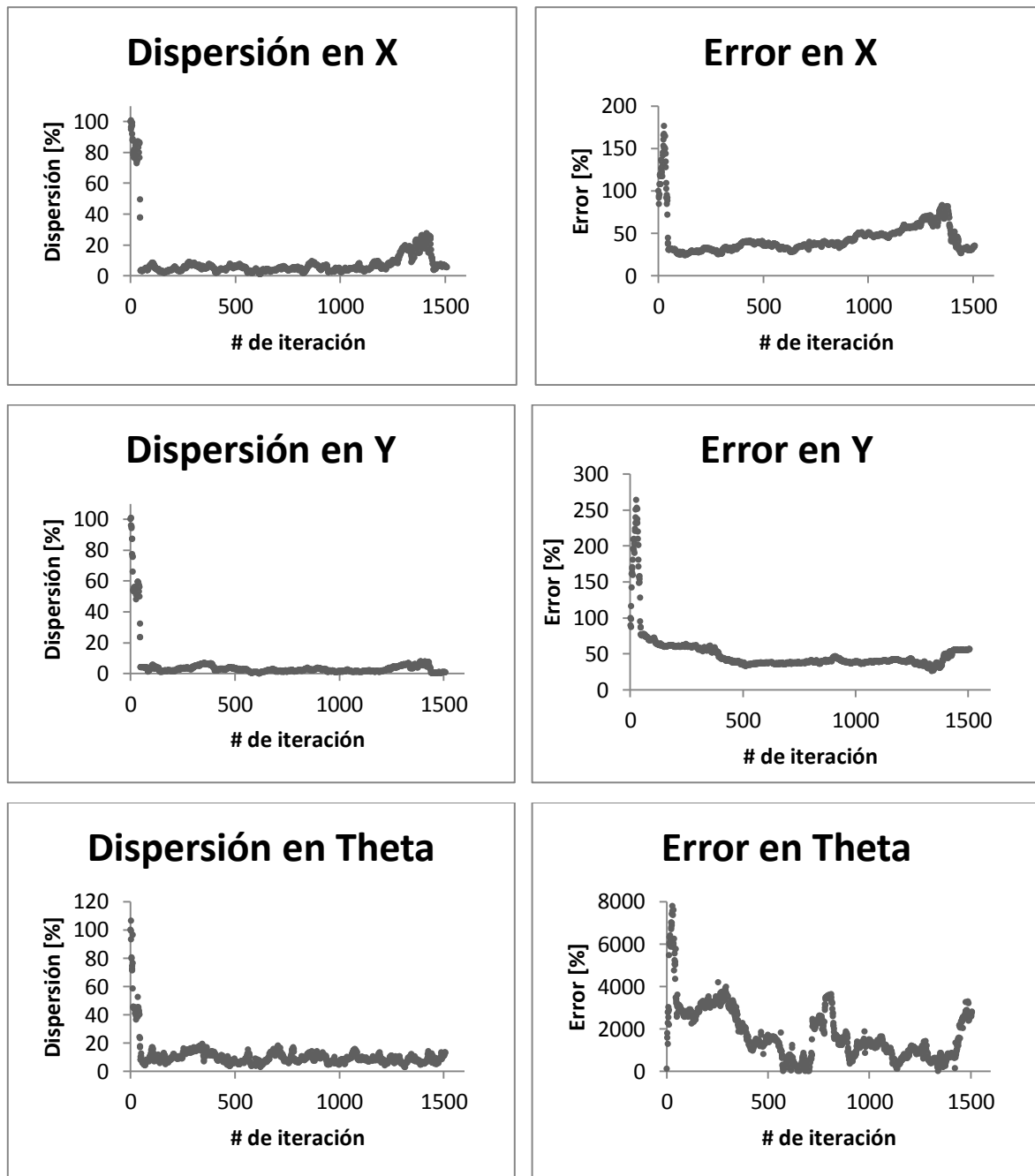


Figura 28. Evolución de la dispersión y el error en cada coordenada del grupo de 100 partículas luego de más de 1500 iteraciones.

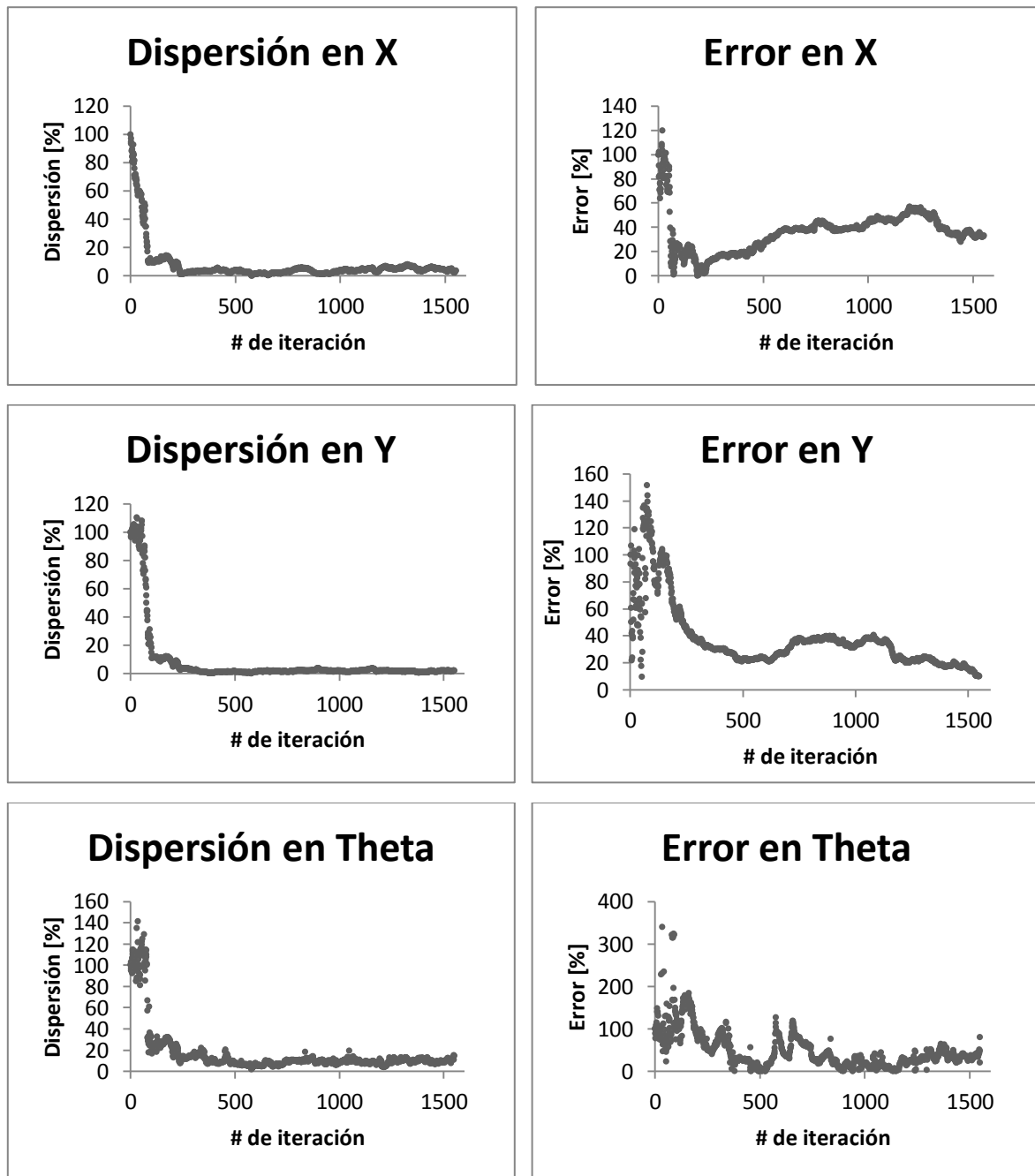


Figura 29. Evolución de la dispersión y el error en cada coordenada del grupo de 500 partículas luego de más de 1500 iteraciones.

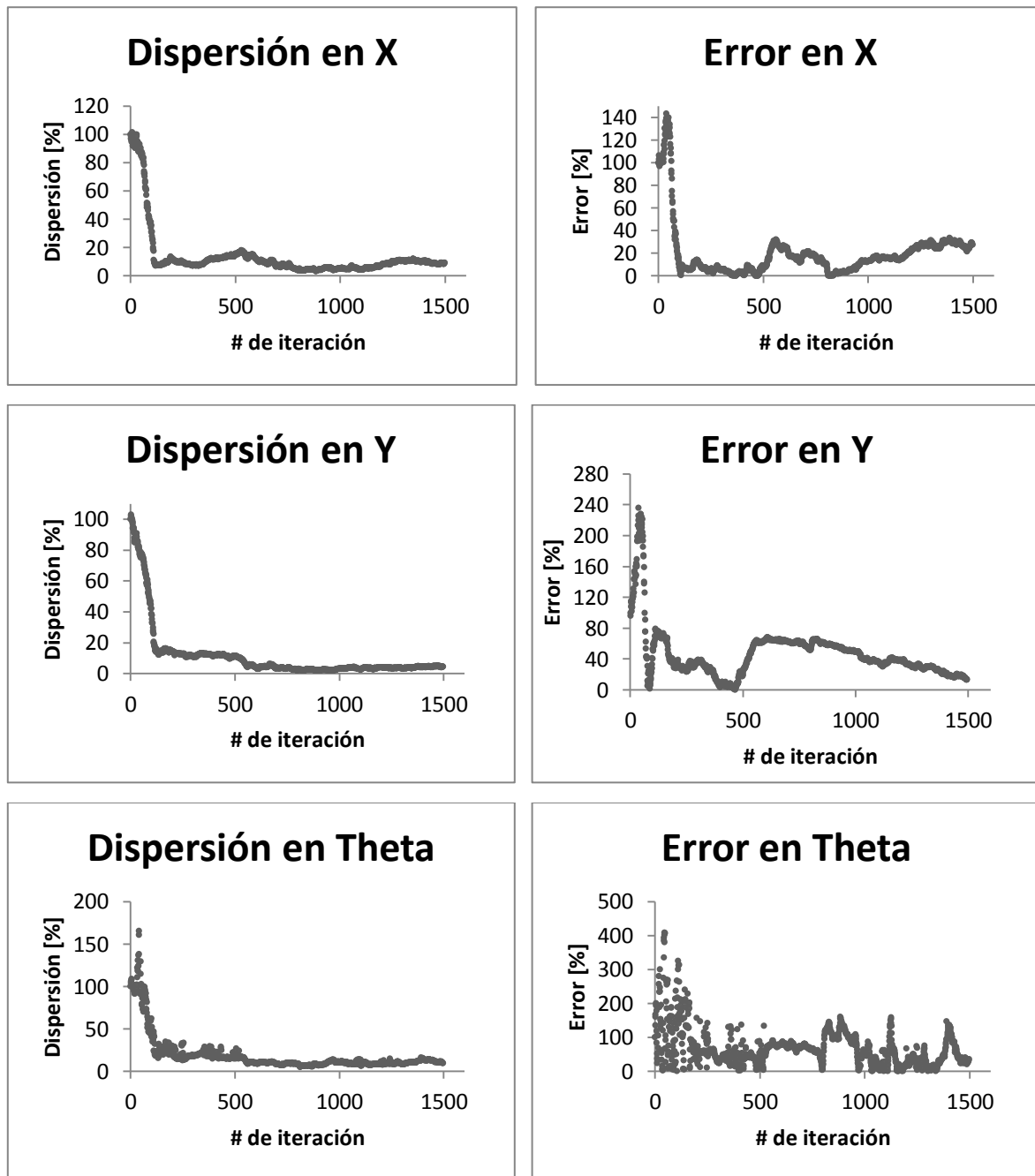


Figura 30. Evolución de la dispersión y el error en cada coordenada del grupo de 1000 partículas.

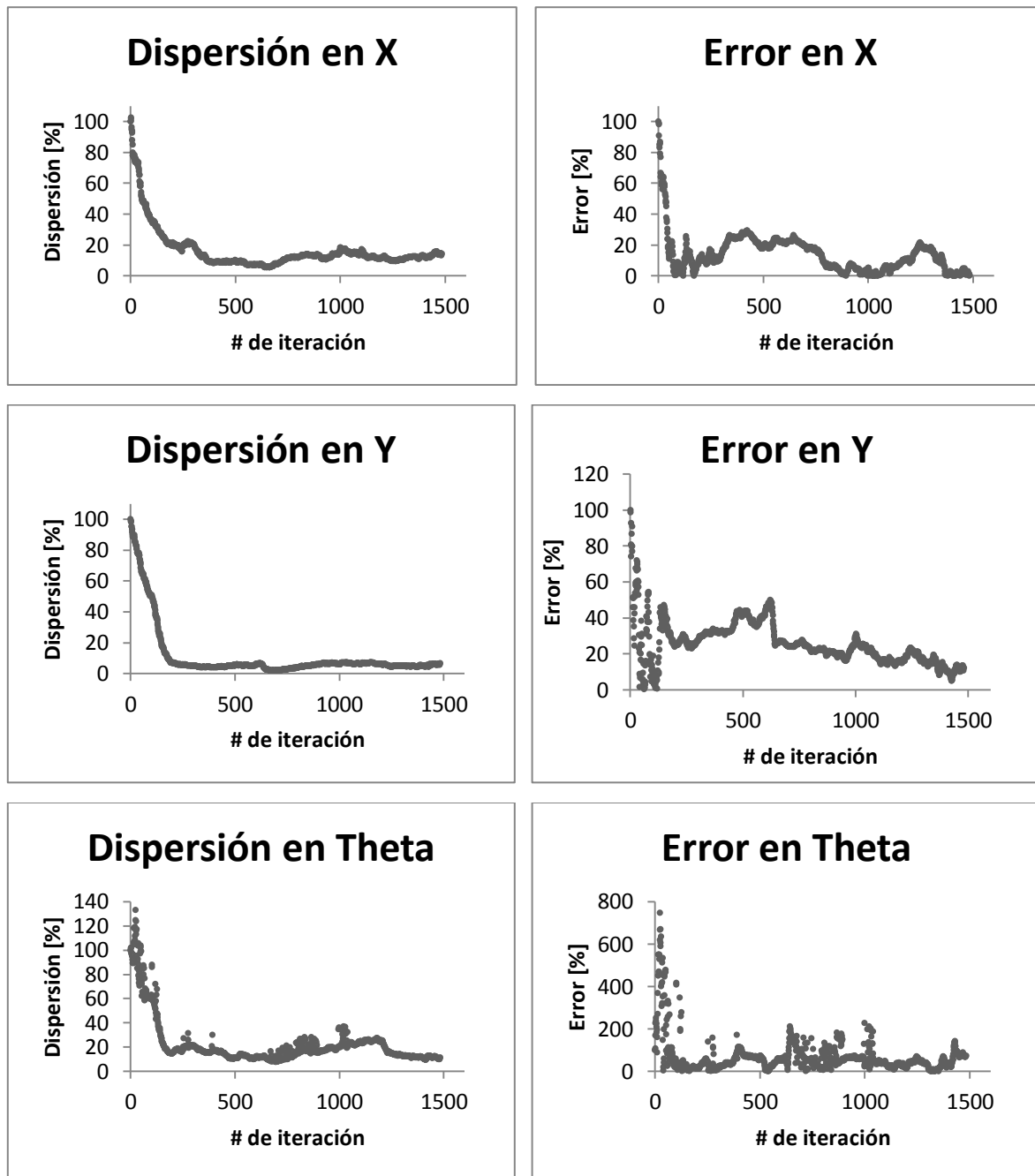


Figura 31. Evolución de la dispersión y el error en cada coordenada del grupo de 1500 partículas.

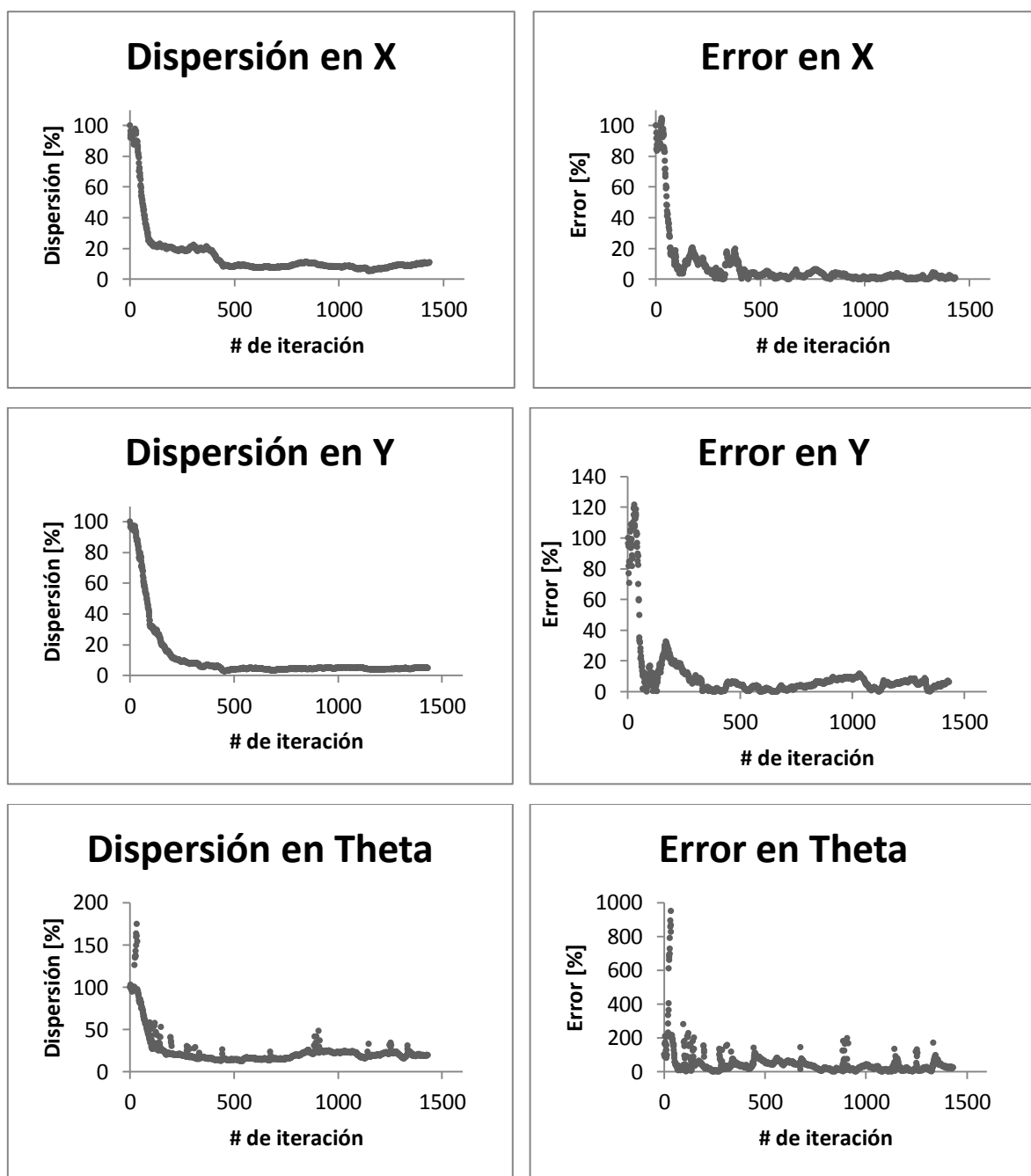
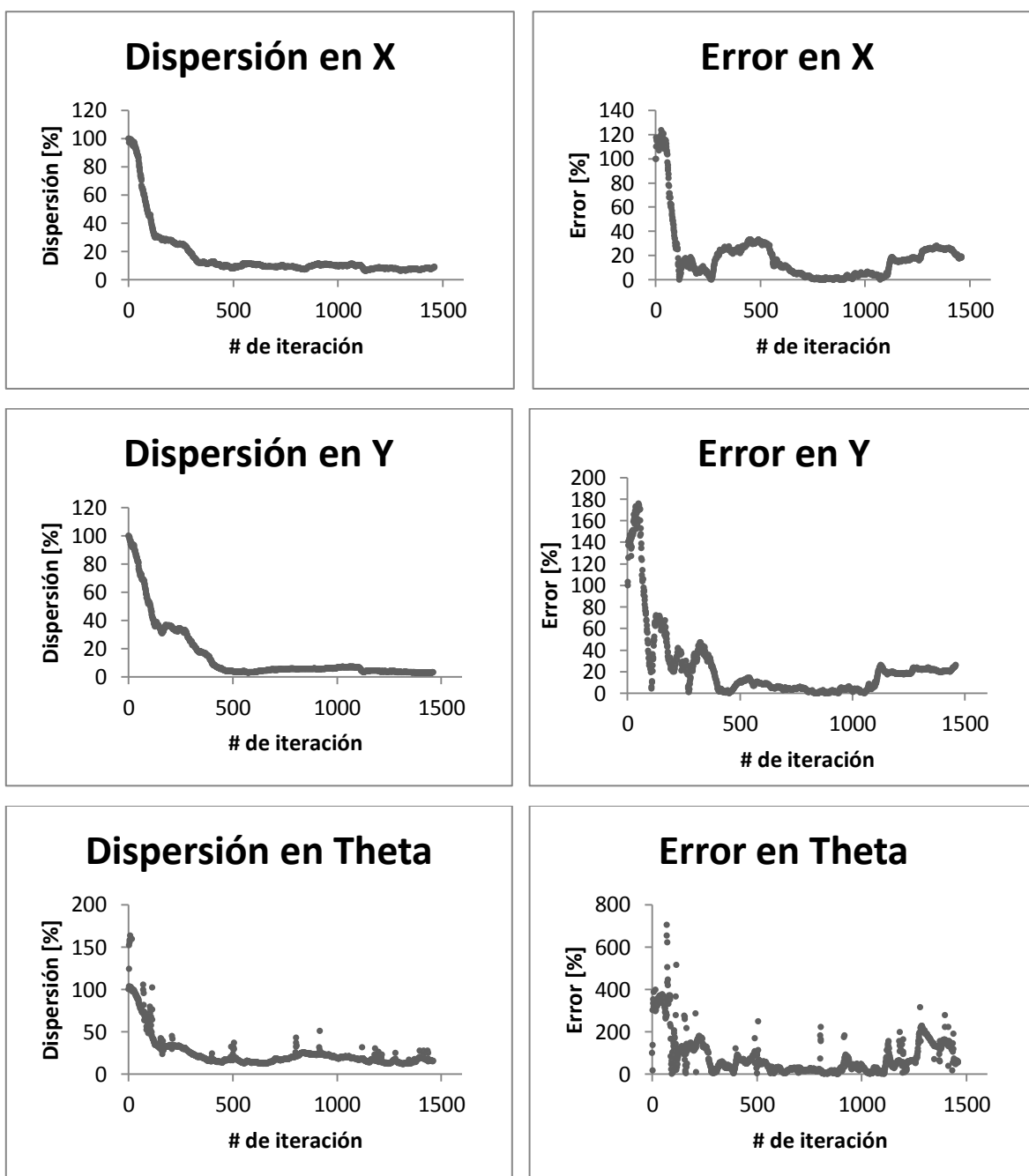


Figura 32. Evolución de la dispersión y el error en cada coordenada del grupo de 2000 partículas.



En las gráficas presentadas en ésta sección se puede apreciar que a partir de un número de $n=1000$ partículas los errores se reducen a menos del 40% y las

dispersiones alrededor del 20% con respecto al valor inicial en las coordenadas X e Y . En el caso de la coordenada $Theta$, a partir de $n=1000$ partículas, el error siempre fue menor o igual al 200%, mientras que la dispersión siempre fue menor al 50%.

Se confirma lo encontrado en la literatura con respecto a la relación convergencia/No. de partículas, en cuanto a que se necesita un alto número de partículas para lograr cierto grado de convergencia a medida que las iteraciones se van dando. Lo anterior se aprecia específicamente al analizar las gráficas de error de cada coordenada.

El comportamiento del error de la coordenada $Theta$ no representa las expectativas que se tenían, ya que se esperaba un comportamiento estrechamente relacionado al de los errores en las coordenadas X e Y .

5.4 SIMULACIÓN DEL ALGORITMO ELCSM-UV VARIANDO LA POSICIÓN DE INICIO DEL ROBOT OBJETO DE LA ESTIMACIÓN DE LOCALIZACIÓN

Luego de simular el algoritmo ELCSM-UV variando el número de partículas, se ejecuta el mismo variando la posición de observación inicial del robot R2 (azul). A continuación se presentan las posiciones y los resultados correspondientes luego de realizadas las simulaciones. El número de partículas implementadas fue de $n=1000$, este valor fue seleccionado de acuerdo al rendimiento obtenido en las pruebas realizadas en la sección 5.3.

Figura 33. Robot R2 ubicado en $x = 0 \text{ m}$, $y = -6 \text{ m}$ y $\theta = 0^\circ$.

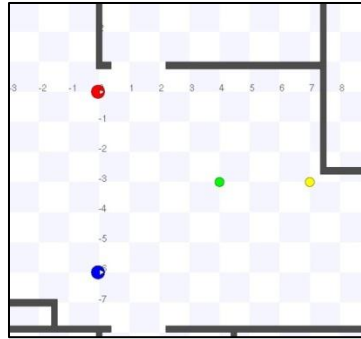


Figura 34. Robot R2 ubicado en $x = -6 \text{ m}$, $y = -6 \text{ m}$ y $\theta = 0^\circ$.

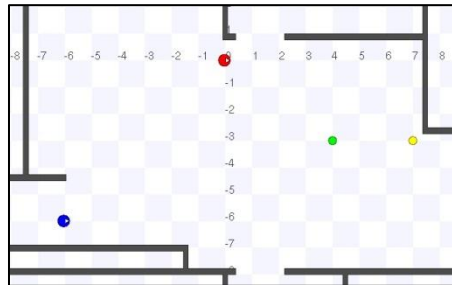
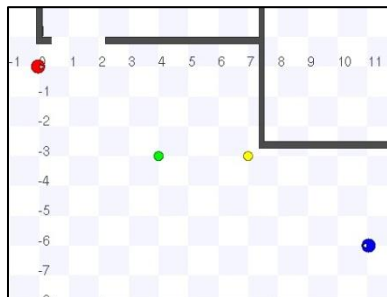


Figura 35. Robot R2 ubicado en $x = 11 \text{ m}$, $y = -6 \text{ m}$ y $\theta = 180^\circ$.



A continuación se presentan las gráficas de los datos recopilados al simular los entornos mostrados de la Figura 33 a la Figura 35. En cada gráfico se representan porcentualmente la dispersión y el error para cada coordenada con respecto a su valor inicial en la iteración uno.

Figura 36. Dispersión y error para cada coordenada de posición del Robot R2 partiendo de la posición $x = 0$, $y = -6$ y $\theta = 0^\circ$.

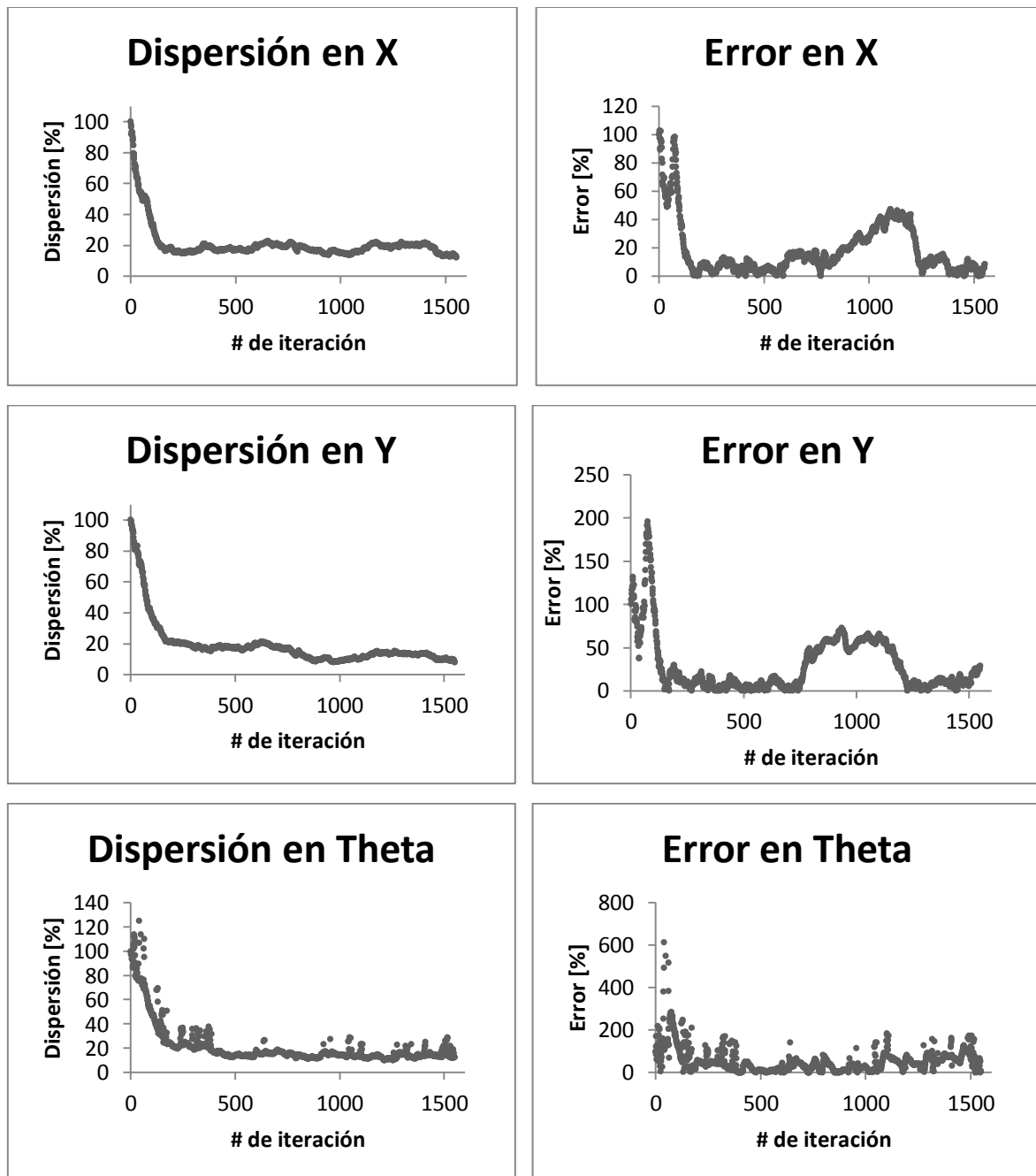


Figura 37. Dispersión y error para cada coordenada de posición del Robot R2 partiendo de la posición $x = -6$, $y = -6$ y $\theta = 0^\circ$.

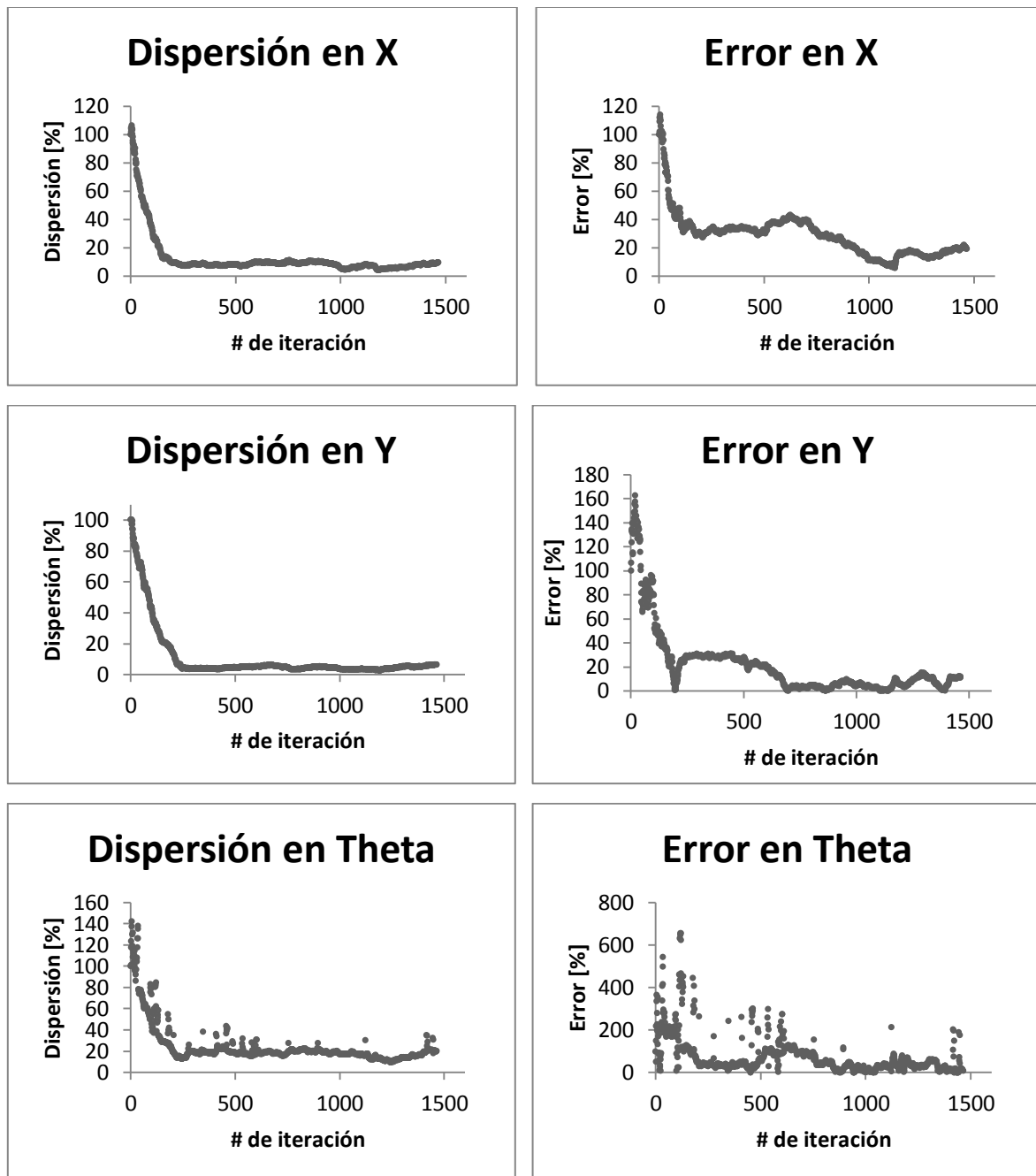
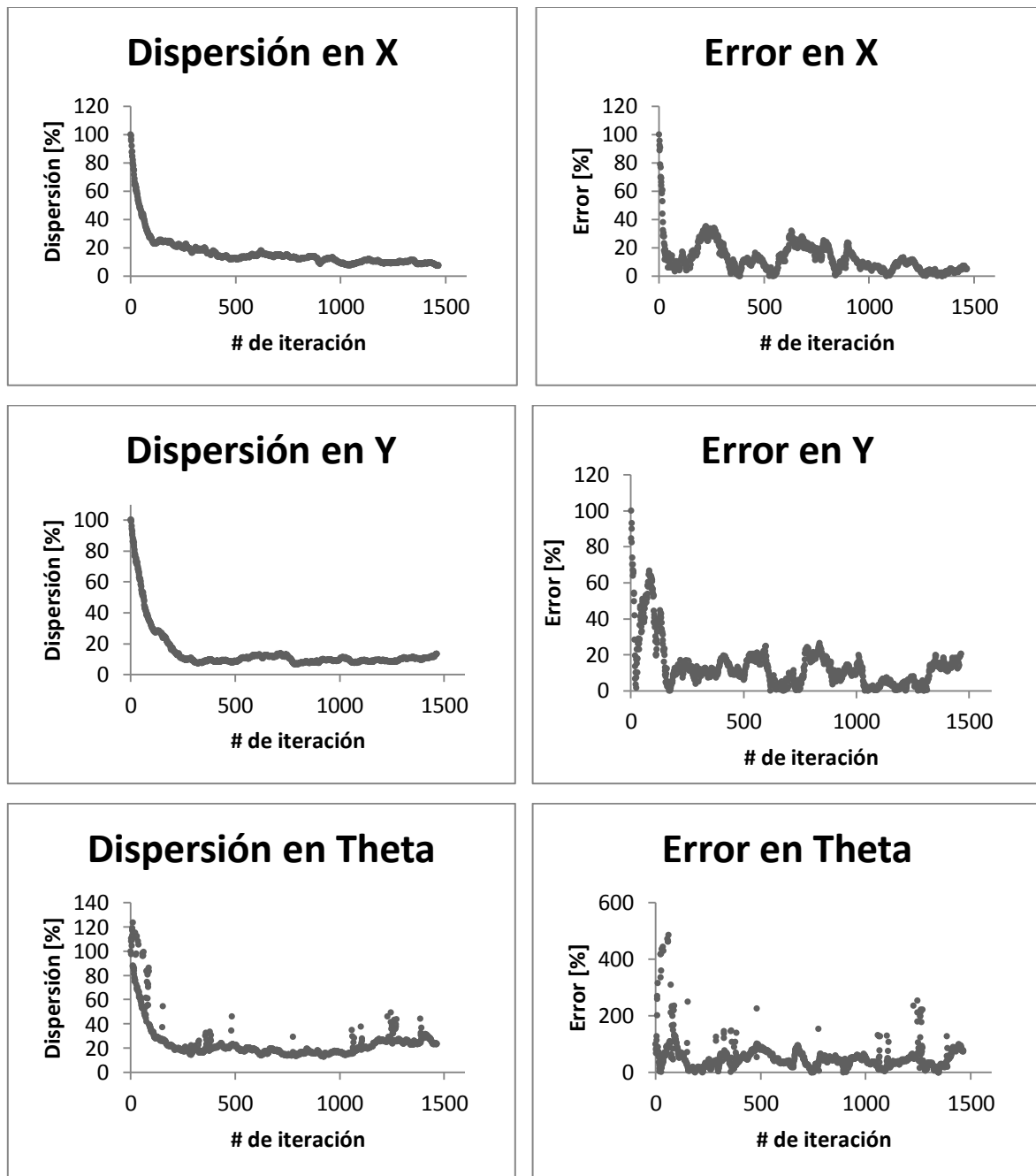


Figura 38. Dispersión y error para cada coordenada de posición del Robot R2 partiendo de la posición $x = 11$, $y = -6$ y $\theta = 180^\circ$.



En las Figuras se muestran los resultados de pruebas al modificar la posición del robot Nx2 (azul) con un número de partículas $n=1000$, generando resultados

similares a los obtenidos en la sección 5.3 con dispersiones menores o iguales al 20% y errores menores al 40% en las coordenadas X e Y ; de igual forma para el caso de la coordenada θ el error fue menor de 200%, mientras que la dispersión de los datos fue menor de 40%.

5.5 CONCLUSIONES

El algoritmo individual presenta mayor dispersión y mayor error cuando los vectores imaginarios entre el robot y las marcas observadas, presentan un ángulo agudo cercano a cero.

El aumento del número de partículas implementadas para representar la estimación de posición de un robot por parte de otro, reduce tanto la dispersión como el error relativo en cada coordenada, esto se aprecia en la sección 5.3 donde el algoritmo con un número de partículas mayor o igual a 1000, logra una reducción de la dispersión de hasta el 20% y del error de hasta el 40% con respecto al valor inicial de la dispersión y del error respectivamente.

Con base en los resultados obtenidos en la sección 5.4 al cambiar la posición del robot al que se desea localizar, los resultados siguen el mismo patrón de comportamiento de los resultados obtenidos en la sección 5.3, dando cierto nivel de confianza al momento de ser implementado.

CONCLUSIONES

Se desarrolló un algoritmo que permite a un robot, estimar la posición de un compañero de equipo de manera indirecta al compartir información recopilada, basado en sus propios sistemas exteroceptivos.

Se recopiló información relacionada con los sistemas multirobot; con la localización en la robótica, métodos y sistemas; y con la localización cooperativa con la intención de fortalecer los conocimientos relacionados con el tema particular de éste trabajo de grado.

Se estableció un algoritmo que permite generar una estimación de la posición de un robot con respecto a referencias externas, a las que se les definió un área de probabilidad que permitieron generar el área de probabilidad o incertidumbre de la posición del robot que las observa.

Se diseñó y desarrolló una estrategia de localización que reduce la incertidumbre generada por el sistema de localización individual, basada en filtro de partículas. Se escogió el filtro de partículas como método probabilístico para reducir la dispersión de las estimaciones de posición, ya que, puede eliminar el ruido de los sistemas con base en la estimación de un estado próximo usando señales de control, mediciones y posiciones actuales. Además, en la Escuela de Ingeniería Eléctrica y Electrónica no se encontraron trabajos de grado en robótica donde se trabajara con este tipo de filtro.

Con la implementación de la estrategia de localización propuesta en este trabajo se obtuvieron dispersiones de datos de menos del 20% y errores menores al 40% con respecto a sus valores iniciales.

TRABAJOS FUTUROS

Como trabajo futuro se propone ampliar el número de robots que ejecutan el algoritmo *ELCSM-UV*, esto con la intención de analizar si al sobre poner las distintas estimaciones de posición de un robot por parte de más de uno de sus compañeros, tanto la dispersión como el error son reducidos aún más.

Implementar la estrategia desarrollada en un sistema multirobot con tareas preestablecidas que requieran apoyo en un sistema de localización de cada uno de los miembros.

Utilizar canales de comunicación en el ambiente simulado para determinar cómo se ve afectada la comunicación del sistema multirobot al escalar el número de sus integrantes.

Llevar la estrategia diseñada a un ambiente real, reportando resultados para evaluar su comportamiento.

Analizar el efecto que produce la incertidumbre de los sensores reales a la estrategia de localización propuesta y analizar la incertidumbre final de la estimación de posición del robot objeto de localización.

BIBLIOGRAFÍA

- [1] John J. Leonard and Hugh F. Durrant-Whyte, "Mobile Robot Localization by Tracking Geometric Beacons", IEEE Transactions on Robotics and Automation, Vol. 7, No. 3, June 1991.
- [2] G. Zecca, P. Couderc, M. Banâtre y R. Beraldi, "Cooperation in swarm of robots using RFID landmarks", IEEE International Workshop on Robotics and Sensors Environments. Ottawa, Canada, 17-18 October 2008.
- [3] Joseph Knuth and Prabir Barooah, "Distributed collaborative localization of multiple vehicles from relative pose measurements", Forty-Seventh Annual Allerton Conference, Allerton House, UIUC, Illinois, USA, September 30 – October 2, 2009.
- [4] L. Diaz. Ambrona-Tabernilla, "Sistema de Localización en interiores", Tesis, Universidad Politécnica de Madrid.
- [5] J. R. Martinez de Dios, "Fusión sensorial mediante Filtro de Kalman con diagnosis de sensores", Notas de clase, Grupo de Robótica, Visión y Control, Dpto. Ing. Sistemas y Automática, Universidad de Sevilla.
- [6] Stergios I. Roumeliotis, "Robust Mobile Robot Localization: from Single-robot Uncertainties to Multirobot Interdependences", Ph.D. Thesis, University of Southern California, May 2000.

- [7] J. Borenstein and Liqiang Feng, "Measurement and correction of systematic odometry errors in mobile robots", IEEE Transaction on Robotics and Automation, Vol. 12, No. 5, October 1996.
- [8] Nini Juliana Vélez Melo, "Sistema de robots móviles cooperativos de realidad aumentada", Universidad del Valle, Colombia, 2009.
- [9] Breyner Posso Bautista, "Selección de una arquitectura de control para un conjunto de robots móviles cooperativos basada en sistemas multiagente", Universidad del Valle, Colombia, 2009.
- [10] J. Hernández and A. Pustowka, "Formation control of cooperative robots with limited sensing using a virtual robot as reference", Latin American Robotics Symposium, September 11, 2009.
- [11] Eval Bladimir Bacca, "Diseño e implementación de un conjunto cooperativo de robots móviles", 2003.
- [12] J. Jiménez, D. Ovalle y J. Ochoa, "SMART: Sistemas multi-agente robótico", Universidad Nacional de Colombia, Medellín, 2007.
- [13] E. González, Laura Cortés, Fabián Roldán, "Robótica cooperativa: Plataforma educativa QUEMES", Universidad Javeriana – Maloka, Campus Party 2012, [Online], Disponible en: <http://www.youtube.com/watch?v=B7uakZ4dJ5s>, Colombia, 2012.
- [14] P. Lima and L. Custodio, "Artificial intelligence and systems theory: applied to cooperative robots", International Journal of Advanced Robotic Systems, Volume 1, Number 3, 2004.

- [15] A. Botía, D. Gallardo y M. Alfonso, "Comportamientos coordinados en formaciones de robots usando percepción visual y comunicación punto a punto", Universidad de Alicante, Alicante.
- [16] L.S. Coelho and M. F. M. Campos, "Pose estimation of autonomous dirigibles using artificial landmarks", Universidade Federal de Minas Gerais.
- [17] R. Kurazume and S. Nagata, "Cooperative position in with multiple robots", Fujitsu Laboratorios LTD, Tokyo Institute of Technology.
- [18] E. Rubión, "Diseño de un algoritmo de localización para un robot móvil".[Online]. Disponible en: http://www.lleida.org/aeti_acces/doc/PDF3/Eng_tecnica/10.PDF.
- [19] "Implementación del FKE para Slam, 2. El problema de la localización y el mapeo simultáneo", Proyecto fin de carrera, Universidad de Sevilla, España.
- [20] I. Navarro Oiza, "Localización de un robot móvil con el filtro extendido de Kalman", 14 de julio de 2005.
- [21] D. Navarro, F. García y G. Benet, "Auto-Localización de robots móviles usando el filtro de Kalman: Un caso basado en hitos puntuales extraídos con sonar", Artículo.
- [22] C. Agüero, J. Cañas, H. Montes y M. Armada, "Trabajando en equipo: Un repaso a los robots móviles coordinados". Proceedings of RoboCity2030 5th Workshop, Cooperación en robótica, pp 11-44, Móstoles, 6 Febrero 2009.

- [23] T. Balch and R. C. Arkin, "Behavior-based Formation Control for Multirobot Teams", IEEE Transactions on Robotics and Automation, Vol. 14, No. 6, December 1998.
- [24] J. Borenstein, H.R. Everett and L. Feng, "Where am I?, Sensors and Methods for Mobile Robot Positioning", Oak Ridge National Lab D&D Program – University of Michigan, 1996.
- [25] A. Ramirez-Serrano, "Navigation of Unmanned Vehicles Using a Swarm of Intelligent Dynamic landmarks", IEEE Workshop on Safety, Security and Rescue (SSRR), June 6-9, 2005, Kobe, Japan.
- [26] A. Barry, "Errors in Practical Measurement in Surveying, Engineering, and Technology", Cap. 7, Landmark Enterprises, 1991.
- [27] S. Thrun, W. Burgard, D. Fox, "Probabilistic Robotics", The MIT Press, 2006.
- [28] D. Zhuo-hua, F. Ming, C. Zi-xing, YU Jin-xia, "An adaptative particle filter for mobile robot fault diagnosis", Journal of Journal of Central South University, 2006.
- [29] Jeong Woo, Young-Joong Kim, Jeong-on Lee and Myo-Taeg Lim, "Localization of Mobile Robot using Particle Filter", SICE-ICASE International Joint Conference, Busan, Korea, October 2006.
- [30] S. Thrun, "Particle Filters in Robotics", Proceedings of the Eighteenth Conference on Uncertainty in Artificial Intelligence, Alberta, Canada, August 1-4 2002.

- [31] G. Cen, N. Matsuhira, J. Hirokawa, H. Ogawa, I. Hagiwara, "Mobile robot global localization using particle filters", International conference on control, Automation and Systems, 2008.
- [32] S. Thrun, W. Burgard, and D. Fox. Probabilistic Robotics. MIT Press, Cambridge, MA, 2005.
- [33] S. Thrun, D. Fox, W. Burgard, F. Dellaert, "Robust Monte Carlo localization for mobile robots", Journal of Artificial Intelligence, 2001
- [34] D. Fox, "Adapting the sample size in particle filters through KLD-sampling", The International Journal of Robotics Research, 2003.
- [35] D. Simon, "Optimal state estimation Kalman , H_∞ and nonlinear approaches", Jhon Wiley and Sons, Inc, 2006.
- [36] M. Sanjeev Arulampalam, S. Maskell, N. Gordon, and Tim Clapp, "A tutorial on particle filters for online non-linear/non-gaussian Bayesian tracking", IEEE Transactions on signal Processing (S1053-587X) 50(2), 174-188, 2002.
- [37] Y. Ling-li, "Fault detection and identification for dead reckoning system of mobile robot based on fuzzy logic particle filter", Journal Central South university, China, 2012.
- [38] A. Gil, O Reinoso, L. Payá y M. Ballesta, "Influencia de los parámetros de un filtro de partículas en la solución al problema de SLAM", Universidad Miguel Hernández, Departamento de Ingeniería de Sistemas Industriales, Alicante, España, 2007.

- [39] R. Havangi, M. Ali Nekoui and M. Teshnehlab, "A Multi Swarm Particle Filter for Mobile Robot Localization", International Journal of Computer Science Issues, Vol. 7, Issue 3, No 2, May 2012.
- [40] S. I. Roumeliotis and G. A. Bekey, "Collective localization: A distributed Kalman filter approach to localization of groups of mobile robots", Proceedings of the 2000 IEEE, International Conference on Robotics & Automation, San Francisco, CA – April 2000.
- [41] S. I. Roumeliotis and G. A. Bekey, "Distributed multirobot localization", IEEE Transactions on Robotics and Automation, Vol. 18, No. 5, October 2002.
- [42] C.E. Agüero, J. M. Cañas, V. Matellán y F. Martín, "Estimación de objetos con fusion Bayesian en equipos de robots móviles", Universidad Rey Juan Carlos, Madrid, España.
- [43] Sv. Noykov, A. Lekova and O. Manolov, "Relative location and cooperation of two mobile vehicles with web-based application", Bulgarian Academy of Sciences, Bulgaria.
- [44] D. Fox, W. Burgard, H. Kruppa, S. Thrun, "A probabilistic approach to collaborative multirobot localization", Autonomous Robots, 8(3), 2000
- [45] M. D. P. Moratuwage, W. S. Wijesoma, B. Kalyan, J. F. Dong, "Collaborative multi-vehicle localization and mapping in marine environments", Massachusetts Institute of Technology, Cambridge, MA.
- [46] W. Navidi, "Statistics for engineers and scientists – 2 edition", McGraw-Hil, 2008.

[47] Sebastian Thrun, Udacity website. CS373 Artificial Intelligence, Programming a Robotic Car Unit 3.20.[Online]. Available: <http://www.udacity.com/>

[48] F. Huerta, “Comparativa de Métodos de Muestreo en un Filtro de Partículas Aplicado al Seguimiento de Múltiples Objetos”, Reporte de Doctorado, Universidad de Alcalá, 2007.

[49] Daniel J. Vimont, “The Bivariate Normal Distribution”, Available: http://www.aos.wisc.edu/~dvimont/aos575/Handouts/bivariate_notes.pdf, Notes, University of Wisconsin – Madison.

[50] B. Gerkey, R. Vaughan and A. Howard, “The Player/Stage Project: Tools for Multirobot and Distributed Sensor Systems”, Proceedings of the International Conference on Advanced Robotics (ICAR 2003), Coimbra, Portugal, 2003.

GLOSARIO

CSLAM: Collaborative Simultaneous Localization and Mapping

EKF: Extended Kalman Filter

KF: Kalman Filter

FP: Filtro de Partículas

GPS: Global Positioning System

IDL: Intelligent Dynamic Landmarks

SLAM: Simultaneous Localization and Mapping

SMR: Sistema Multirobot